

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号
特開2001-357190
(P2001-357190A)

(43)公開日 平成13年12月26日 (2001. 12. 26)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード*(参考)
G 0 6 F 17/60	1 6 8	G 0 6 F 17/60	1 6 8
19/00	1 1 0	19/00	1 1 0

審査請求 未請求 請求項の数29 O L (全 31 頁)

(21)出願番号 特願2001-110877(P2001-110877)
(22)出願日 平成13年4月10日(2001. 4. 10)
(31)優先権主張番号 特願2000-111033(P2000-111033)
(32)優先日 平成12年4月12日(2000. 4. 12)
(33)優先権主張国 日本 (J P)

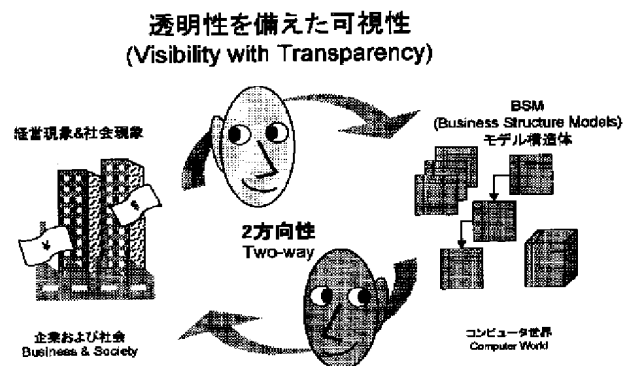
(71)出願人 300032813
アヴィクス株式会社
東京都杉並区西荻北3丁目14番18号
(72)発明者 東原 功
東京都杉並区西荻北3丁目14番18号 アヴ
ィクス株 式会社内
(72)発明者 外山 ▲たか▼之
神奈川県川崎市宮前区宮崎3丁目5番地26

(54)【発明の名称】 B S Mの企画・設計・構築・保守及び運用に関する方法

(57)【要約】

【目的】 経営や社会事象をその要素間で最大限総当たりの因果を持つ関係と捉え、多階層、多次元の構造体からなり、情報処理装置にて演算可能なモデルとして写像し、このモデルから、逆に事象を想像できるという双方向の透明性を備えた可視性あるモデリング(B S Mモデル)を、式に依存することなく、表または表群で実現し、これを操作して、意図した条件に対する因果分析と将来の適切な行動を探る仕組みを提供する。

【構成】 本発明は、インプット、プロセス、アウトプットの3領域を持ち、その構成要素が演算関係や共有構造を持つハウジングを基本として、多階層のハウジングの入れ子構造を形成し、かつハウジングの収納力所であるルームに多様な形態のデータを収納し、時系列モデルを形成し、モデル間の多様な連鎖を形成し、開発・運用とモデル演算によるシミュレーションをエンドユーザの対話操作が可能な構造である。



【特許請求の範囲】

【請求項1】BSMの対象となる企業の組織や活動の事象（機能、時空、知識およびこれらの関係）を構成する要素を抽出し、これら事象を上辺部と左辺部の要素の間で共有する総当たりを表す関係テーブル形式でBSMモデルを構築する方法であって、この関係テーブルに、上辺部、左辺部、中央部の関係、中央部の要素およびその処理結果が上辺部に遷移する機能からなる一単位のBSMモデルを構築し、これらのBSMモデルを入れ子の構成要素とするBSM多階層構造体モデルを構築すること

10

を特徴とするBSMの企画・設計・構築・保守及び運用に関する方法。

【請求項2】（ハウジングと入れ子）モデリングの基本単位を演算性ある因果関係を表すハウジングとし、これに別のハウジングが入れ子を成す（以降、入れ子ハウジングという）ことにより、BSMの階層構造を形成し、これを組み替えて多様な対象に対話型操作で対処できるモデル構造体を形成し、これを演算実行して多様なケース・スタディの結果を得ると同時に個々のハウジングは複数のハウジングに入れ子となりえ、モデリングの対象

20

の分担とコラボレーションを行える構造とする方法。

【請求項3】（ハウジングの構造）格子棚状のルームからなるハウジングは、上辺部、左辺部、および中央部の3区域からなり、上辺部、左辺部のルームへは、テナントあるいは、複数のテナントをカプセル化した認識記号を持つ、カプセル・テナントを配置し、これら名称は、上辺部および左辺部では、異綴の名称である制約を持ち、かつ、両部では同綴の名称を共有できる構造で、中央部のルームには空テナントを含め、テナント、ある

30

いはカプセル・テナントを配置し、テナントは同綴名も可能であり、テナントとカプセル・テナントを、あたかも同等に扱うハウジングの構造と運用の方法。

【請求項4】（ハウジングにおけるテナントへのアクセス構造）ハウジングのルームに格納されるテナントあるいはカプセル・テナントは、そのヘッダー情報とボディ情報からなり、ヘッダー情報の一部を、ハウジングのルームの配置を反映して画面上のルーミング・テーブルに登録し、ボディ情報が空でない（実在する）場合は画面に展開し設定する、あるいは変更の画面に展開して作業し、また別時点でルーミング・テーブルをアクセスしてテナントあるいはカプセル・テナントの情報にアクセスでき、このテーブルからテナントおよびそのボディ情報の追加、削除、挿入、名称変更の操作を行える構造を持つハウジング構造とその運用の方法。

40

【請求項5】（ハウジングの有効メインライン演算域）ハウジングを縦方向に見て、ルームにテナント／カプセル・テナントが上辺部あるいは中央部に配置されていない列領域、または横方向に見て、ルームにテナント／カ

50

プセル・テナントが左辺部あるいは中央部に配置されていない行領域を除くハウジングの領域を、有効メインライン演算域とするハウジングの構造。

【請求項6】（基本メインライン演算およびメインライン演算）ハウジングの有効メインライン演算域において、上辺部に存在し左辺部に存在しないテナント／カプセル・テナントを先決テナント／カプセル・テナント、両辺部に存在するテナント／カプセル・テナントを中間テナント／カプセル・テナント、左辺部にのみ存在するテナント／カプセル・テナントを後決テナント／カプセル・テナントとし、

ハウジングの基本メインライン演算は、先決テナント／先決カプセル・テナントを出発候補として、上辺部、中央部、左辺部のテナント／カプセル・テナントの組み合わせで決まる演算を経て、中間テナント／カプセル・テナントに結果を得て、全ての有効メインライン演算域を巡り、後決テナント／後決カプセル・テナントに結果を得るもので、演算シーケンスをハウジングの配置から自動的に確立する。さらに、基本メインライン演算を含み、その結果を、先決あるいは中間結果のテナントに戻し、ループ演算を、特定のテナントの値が満足を得られるまで、あるいは、うち切り限度まで演算するハウジングを巡るメインライン演算の方法。

【請求項7】（テナント情報）ハウジングのルームにはエンドユーザが意識しやすい一覧して識別できるテナント／カプセル・テナント、演算で使用する情報およびその運用を管理し促進する多様な形態で格納するもので、中央部のルーム収納情報には、テナント／カプセル・テナント名の一部に基本メインライン演算を指定する文字列おも含む構造とする。

【請求項8】（繰込・繰出情報）テナント／カプセル・テナントのボディとなる情報は、メインライン演算に使用する、あるいはその中間結果を扱う段階、その事前準備の段階、メインライン演算後の事後段階、演算・運用の複数の段階に応じて、取り扱いとそれに応じた扱いの段階を分けて収納する構造とする。

【請求項9】（項目リスト）テナント／カプセル・テナントが、スカラーを含む多次元テーブルである場合、その要素個々の意味付けと位置認識のために、項目リスト名を代表名とし、項目相互が非同綴の文字列のからなる項目リストを、表示、変更操作、外部との入出力、メインライン演算に、その軸情報として随伴させて扱うことを可能とし、項目リストはハウジング内で共有できる構造とする。

【請求項10】（不確定項目を含む項目リスト）メインライン演算および外部情報の読み込みの結果、項目リストの項目が変化する場合の項目リスト名には、エンドユーザおよびシステムが認識しやすい記号を頭部に付して扱い（以降、キャッピングという）、演算にも対処する方法。

【請求項11】(項目リスト内の欄指定)個々の項目リスト内において、項目の特定位置の文字列欄を特定欄として一括した名称で扱える方法。

【請求項12】(多次元テーブルの軸および随伴する項目リストの対応)多次元テーブルを扱うテナント/カプセル・テナント相互の取り扱いにおいて、統一した次元軸方向を扱い、随伴する項目リストを省略した場合は、対応軸に随伴する項目リストを使用し、また対応する軸が異なる項目リストを持つ場合には、双方の項目リストを照合して、テナント/カプセル・テナントのボディの要素の位置整合を行って、演算を行い結果を収納する。

【請求項13】(テナント間演算)中央部のルームに登録するテナント名の選び方によって基本メインライン演算における縦方向演算の演算方法を規定し、並列する縦方向演算が存在する場合は、その横方向演算についても規定する規則を定めて、テナント名の先頭文字列が演算関数を引き出し、上辺部の直上テナントを使用する演算、および、該当する中央部のルームに収納のテナント情報名の一部で直接、あるいは間接に指定する直上以外の上辺部のテナントを指定して、複数以上の上辺部テナントおよび中央部の該当テナントが抱える収納情報を用いて、これらテナントが使用できる時点で基本メインライン演算し、中央部の横方向のテナントに全ての中間結果が揃う時点で、横方向演算を行い、左辺部のテナント/カプセル・テナントに結果を得ることを可能とする。

【請求項14】(演算種別の登録と設定)演算種別の内には、下位の入れ子ハウジングと上位ハウジングのテナントとの情報の受け渡し、下位ハウジングの上位ハウジングからのメインラインの演算実行をも含むものとする。

【請求項15】(エン・カプセルおよびデ・カプセル演算)テナントをカプセル化してカプセル・テナントにするエン・カプセル演算、逆に、カプセル・テナントからテナントを取り出すデ・カプセル演算を可能とする。

【請求項16】(ハウジング間およびカプセル・テナントーテナント間受け渡し演算)上位ハウジングの傘下で、下位ハウジング間で、あるいはカプセル・テナントとテナント間でそれらテナントの要素間の受け渡しをする演算を可能とする。

【請求項17】(ミニ・ハウジング演算)演算種別の形態として、ハウジングの各テナントが一要素で構成される集合領域について、これを上辺部、左辺部、中央部の各一個のテナントで演算できる形態を可能とする。

【請求項18】(モデル領域の一括複写)ハウジングのルーミング・テーブルの中央部において、領域を指定し、かつ挿入点を指定することにより、その領域の直上の上辺部、および直左の左辺部を、一括して挿入点以降の対応する区域に同綴を避ける文字指定操作を含めて複写操作を可能とする。

【請求項19】(モデル領域の一括転置・複写)請求項18におけると同様な領域指定により、指定領域を転置し

転置演算文字を付して複写でき、その指定領域の直上の上辺部、および直左の左辺部を、一括して挿入点以降の対応する区域に同綴名を避ける文字指定操作を含めて転置し複写可能とし、中央部の複写後のテナントの演算においては、複写前の対応するテナントのボディのメインライン演算に使用した情報を転置して借用することを可能とする。

【請求項20】(ハウジング直下の入れ子ハウジング)入れ子ハウジングは、その上辺部、中央部、左辺部を、それぞれカプセル・テナントとして、その区域を識別できる記号または文字列を付加して、上位ハウジングのそれぞれ対応の区域に、直上、直左の関係を保って配置し、上位ハウジングの基本メインライン演算のシーケンス確立にあたっては、入れ子ハウジングの配置を認識し、上位ハウジングのメインライン演算シーケンスへの組込み演算を確立し、実行する方法。

【請求項21】(入れ子による異なる階層ハウジング間の垂直連鎖)ハウジングのメインライン演算の途中で直下の入れ子ハウジングに、上位のハウジングの情報を渡し、その入れ子ハウジングのメインライン演算をし、その結果の情報を、上位のハウジングに受取り、上位ハウジングのメインライン演算を行う異なる階層ハウジング間を転移する垂直連鎖の方法。

【請求項22】(垂直連鎖の生成促進方法)他のハウジングを入れ子とするためにハウジング・リストから操作者が候補を選び、上位ハウジングの中央部の挿入開始点を指定すると、請求項22による垂直連鎖に関係するテナントまたはカプセル・テナントの設定作業を自動的にを行い、垂直連鎖の登録を促進する方法。

【請求項23】(水平連鎖の生成)上位ハウジングの傘下で、複数のハウジング間を連鎖する(以降、水平連鎖という)場合、ハウジング・リストで水平連鎖の候補となるハウジングを選択し、下位ハウジングのメインライン演算と下位ハウジング間の情報の受け渡しによる水平連鎖を可能とするモデルの自動生成に関する方法。

【請求項24】(水平連鎖の組込)上位ハウジングの直下に水平連鎖を組み込み、上位ハウジングのデータを渡し水平連鎖を実行し、その結果を上位ハウジング内に取り込む、ハウジング間で転移しながら演算を行うモデルの構築と運用に関する方法。

【請求項25】(オカレンス)エンドユーザにとってハウジングが同じ構造の繰返しというイメージで把握しやすく、ハウジング間でルーミング・テーブルを共有化できる場合、これらをオカレンスとし、各々のハウジングは、単一のハウジングとして、上位ハウジングの傘下で、メインライン演算の流れで演算を行う、上位ハウジングの傘下で、待機段階にあるハウジングとしてオカレンス同期情報を持ち、上位ハウジングの設定する同期情報に従ってメインライン演算を行う、あるいはオカレンス内のシーケンスに従って、上位のハウジングに戻るこ

となく別のオカレンスの情報を参照できる構造とする。

【請求項26】(モデル分析からルーミング・テーブルへのモデル設定段階)対象とする現象をモデル化する場合、先決のテナント／カプセル・テナントを上辺部、中間結果を含む結果のテナント／カプセル・テナントを左辺部に寄せ、その後、中央部のテナント／カプセル・テナントを登録するステップを、プロンプト表示を伴う画面操作により実施し、ルーミング・テーブルを構築する方法。

【請求項27】(グローバル領域)グローバル情報として異なる領域(以降、これをグローバル領域という)で保管し、特定のハウジングを越えて、これを共用あるいは借用するために、入れ子として使用するハウジング、テナントおよび項目リストを識別し、変更はこの領域のみで可能な構造とする方法。

【請求項28】ネットワーク環境の機能を活用して行うことを特徴とする請求項1ないし27の何れかに記載のBSMの企画・設計・構築・保守及び運用に関する方法。

【請求項29】BSMモデルの各階層で作成・変更を対話型で可能とし、受け渡し機構の自動変更、および演算順の自動確立に適應できるとともに、演算対象となるBSMの収納情報に随伴する情報の変更についても、影響箇所を予告し、かつ変更に対応して収納情報の記録についても配置上の対応をすることを特徴とし、仕組みの登録、情報の変更、演算の実行フェーズなど種々の段階をパラレルに処理し、メインライン演算のシーケンスおよびその演算タイプと相まって、処理時間の効率化を実現することを特徴とする請求項1ないし28の何れかに記載のBSMの企画・設計・構築・保守及び運用に関する方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータを用い、複雑な企業活動における経営の実績や経営計画の立案、社会および自然の事象の因果関係を、「透明性を備えた可視性」(Visibility with Transparency)かつ現実感をもって表現し、シミュレーションする、BSMの企画・設計・構築・保守及び運用の方法に関する。

【0002】利用分野は、ほとんど全ての、企業において、現業の持つ価値尺度で動く分野と金銭価値で動く分野を明確な因果論理で連鎖する、計画・予算・実績評価管理の業務のモデリングに適用可能であり、生産・原価計画システム、利益計画、事業計画、連結会計システム、給与・昇進計画システム、さらには、社会問題に関しても環境・エネルギー・経済を結ぶモデリングに適用可能である。

【0003】なお、ここにBSMとは、仕組みを指称し、ビジネス・ストラクチャー・モデリング(Business Structure Modelling)の略称である。また、モデル

はBSMモデルという。

【0004】

【従来の技術】従来、これらの分野にアプローチを試みる技法としては、プロシジャー言語、HIPO分析に基づく設計手法、スプレッド・シートおよびストラクチャー・マトリクスがある。

【0005】

【発明が解決しようとする課題】経営の実績計算や経営計画の立案を行うためのモデル作成と運営をプロシジャー言語に基づいて行うには、通常、膨大なステップ数(時として数百万あるいは数千万ステップ、に及ぶ)のコーディング作業を伴い、エンドユーザのみによるモデルの開発と運用は事実上不可能である。この分野のモデルの開発には、全体把握に基づく分析、仕様書の作成、設計、コーディング、テスト、ドキュメンテーション等の一環した開発フェーズを必要とし、プロジェクト開発の専門家群の動員が必要となる。

【0006】このために、開発に時間がかかるだけでなく、開発専門の担当者とエンド・ユーザとの間に意味上の理解、解釈上や価値観の隔たり(セマンティック・ギャップ)が拡大し、意志の疎通が困難となり、設計、コーディングのみならず、結果の検証、変更への対処に多くの時間を要し、その間にニーズが変動して開発そのものの意味がなくなる場合が多い。

【0007】また、情報とプログラムが別々に存在し、情報と演算の有機的な因果関係を把握することは難しい。このため、プログラムから現象を逆に想像することは、システムが大きくなると極めて困難である。

【0008】個々に開発された多数のモジュールを連鎖させるためには、変数の統合、個々のモジュールの実行タイミングを考慮してリンクする必要がある。このため、プログラムの変更・追加が難しい。また、システム開発の観点からもプログラムサイズに限界がある。

【0009】これらの理由により、経営や社会現象あるいはエンド・ユーザの発想とニーズに同期したダイナミックなモデルの開発・変更・維持をプロシジャー言語に基づいて行うことは困難であり、いわゆるコラボレーション(異なる背景を持つ多人数の業務担当者が協同作業をする)によるモデルの開発・運営は一層、および難しい。

【0010】スプレッド・シートは、セルの集計計算を中心とする報告書作成のためのテーブル処理から発達してきたため、基本的にモデル構築とその運用に適したものではない。原理的に各セルの情報の背後にその処理ロジックが隠され、セル単位で扱うことを原則としており、ある領域のセル群に共通の意味を与え、これを纏まった集団として扱い、かつ因果の繋がりを流れとして追いうる構造になっていない。このため、処理ロジックが大きくなるとユーザーにロジックの可視性を欠き、多岐にわたって集約したり分岐したりするセル間の相互関係を一覽して把握できず、複雑に因果関係が絡み合った巨

大な現象をモデルとして把握し、維持するには限界がある。

【0011】また、親子の因果関係、投入産出の因果モデルや生産モデル、組織間配賦など論理は比較的簡単でも要素数が多く、これをネットワーク状に表現した上で、頻繁な項目の増減を扱うテーブルをベースにしたループ処理は、セルに組み込んだ関数処理で処理することは、透明性を備えた可視性の観点からは、程遠い。しかも、よく使い込んだものほど属人的になり、開発者以外の者には適用業務と因果関係を反映した演算の仕組みとの関係を理解しがたくなる。特にスプレッド・シートのマクロを使うと何をやっているか見えなくなりマクロの専門家のみの関与する業務となる例が多い。

【0012】現在のところ、スプレッド・シートには、この分野で使われる巨大なモデル（例えば、縦：1兆セル、横：1兆セルのスプレッド・シート）をモデリングする方法は提示されていない（現在、最も普及したもの最大サイズは、256セル×65,536セルである）。また、一つのセルの入力毎に計算式を入力しなければならず、また一つのセルの情報入力毎に全セルを演算する構造のため、大きなモデルで数百セルを変更し演算する場合は入力設定に時間が掛かりすぎて不能率である。

【0013】スプレッド・シート間で1万件にも亘る情報の受け渡しは、渡すセルと受け取るセル間の詳細な設定とその状況の把握の観点から事実上不可能であり、さらに数百のスプレッド・シート間で同様な受け渡しが発生する場合には、とうてい対応できない。さらに、この受け渡しの関係を演算の必要時点でダイナミックに組み替えて運用するニーズには応えられず、個人向けの作業用途、小企業向けの用途、あるいは大企業の業務を集約した部分、大企業の特定の部分に限定した場合等のモデリングに限られて使用されている。

【0014】上途の2方法をベースに実用モデルを作成する場合には、現実の越えがたい問題（追加すべき機能）が多々ある。一般に、エンド・ユーザはモデリングの経験と共に成長する。そのため、対象とするモデルの広がり、部分から発し、全体へと、学習を伴いながら広がっていく性質がある。つまり、より高位の観点からの上位の階層モデルの形成へ、また、より精巧なモデル化を目指して掘り下げたモデル階層の形成へ、あるいは、同レベルの横広がりを対象としたモデルの形成へと、予測しない多方面への拡張可能性が必要である。このように任意の着手部分から出発して全体形成へ、あるいは詳細部分のモデル形成へとモデリング本来の性格に対する対処の方法が示されていない。

【0015】（HIPO分析に基づく設計手法）従来技術の一つとして、HIPO(Hierarchy plus Input Process Output)に基づく分析、設計の手法がある。これによれば、システム化の事象を木構造の階層を形成すると見なし、各階層毎に入力、出力、およびその間の演算と

いう3要素の関係として、それぞれ入力要素、出力要素、演算要素の域に記述し、相互の関連をHIPOダイアグラムとして表す分析手法がある。この手法によれば、インプット・データ、コンピュータのファイルあるいはデータ・ベースとプログラムの関係を対応させやすいというメリットがあるにもかかわらず、次の理由により、BSMの手法にはなりがたい。

【0016】この手法は、対象とする現象をルートから木構造と見なすため、低位の階層ほど階層の傘を超えた関係を取り扱い難く、下位階層で相互に関連する事象を扱うBSMには不適である。特に、無駄な活動事象の源泉にたどり、原価源泉を追求するBSMの分野でのモデル化のアプローチには、致命的である。

【0017】一般には、この手法にによる記述は膨大な紙数になり、ページをまたがる相互の関係を把握し難く、エンド・ユーザが記述から、実際の経営現象を全体感を持ちながら想像することは、極めて困難である。

【0018】分析した紙上の結果と、システム開発の手法および運用の間が一貫していないためダイナミックな変更要求に追従できない。さらに、変更の結果をドキュメンテーションに反映し続けることが困難である。

【0019】ストラクチャー・マトリクスの名は慣用的に用いられ、経営管理システムの分野でもっとも早く学術文献に現れたのは、1953年であり、確たる定義は存在しないが、その基本は

線形変換式： $A=M \times B$ （Mは正方マトリクス、A,Bはベクトル）

において、 $A=B$ となる値を求める場合、A,Bの要素を区画に分け、この区画に合わせて、Mも区切られた区画とし、区画単位で把握し扱うことを前提に、生産と原価の関係を表現するために考え出されたものである。

【0020】従って、この方法によれば線形構造を持つ領域：製品から製造の流れに逆らって投入資源を求める、投入資源の原価を与え製品の原価を与える、の2つ領域でのモデル化には役だってきた。しかし、広範なBSMの対象領域に対処するには、次の様な問題が存在する。

【0021】BSM事象をインプット域、アウトプット域、プロセス域に分けるなど分かりやすく扱うことができない。

【0022】ストラクチャー・マトリクスにおいては、基本は、上辺と中央のブロック同志の縦積・横和演算あるいはその拡張として上辺部と中央部のブロック間の非線形演算に関する結果を中央ブロックに一時的に蓄え、あらためて横方向の演算として、これら蓄えた結果を加算しその結果を左辺部に収納する。この左辺部に収納する演算は、その定義上で限定し、極めて限られた加算(減算を含む)のみに限定している。本来、広範な四則演算、条件演算、多変数による関数演算などを実行できる構造でないため、複雑な演算の実行性を阻んでいる。さ

らに、計算ステップの大幅な増加で対処しなければならず、モデリングと計算の効率性、システムのメンテナンス性能、ユーザーにわかりやすさを簡潔に提供できないで居る。

【0023】ストラクチャー・マトリクスにおいては、中央部に数値マトリクスを配置し、上辺部をベクトルあるいはマトリクスとして内積が成立しその結果を加えて左辺部の結果とする構造であり、中央部、上辺部、左辺部を区画化してブロック化しても同様に、中央部ブロックと上辺ブロック間で内積が成立し、その中間結果を加えて左辺ブロックの結果とする構造となっている。このため、この枠組みの中で、中央部ブロックに実体を伴わないブロックあるいは演算を指定して、他のブロックの値を借用する、あるいは借用して後、別形態に変換し全体の演算の流れに納めると、という発想がない。

【0024】多ブロック入力要素群の相互の演算を一般化した方式として扱うことができない。非金銭投入価値量と非金銭産出価値量の関係、投入資源の単価と産出単価の関係のみでなく、これらの積である総金額をも合わせて、3系統の事象を1つの領域で扱うことができない。

【0025】実際の経営現象には、線形の関係が成立するケースは少なく、非線形のケースが多い。しかし、ストラクチャー・マトリクスは本来、線形現象を表現する方法として出現した。このためその線形構造を延長して、かつ非線形を主体とする一般的で複雑な事象を主体とするモデルを本格的に扱えない。

【0026】平面的な因果関係の表現から出発したストラクチャー・マトリクスでは現象に合わせて、入れ子による上下両方に任意の多階層を成す事象の発想がなく、表現も不可能で、その構造を組み込めない。

【0027】ストラクチャー・マトリクスは、本来、線形の因果関係を表現し、その関係に沿って演算を実現するためのものである。このため操作者との簡易で分かりやすいブロックに関する入出力を支援したり、上辺ブロックとの計算に先立つ事前の調整計算を組み込む、外部データをその入力場所に即して扱い保管する、多次元のデータや多様な形式のデータを扱い、ブロックをに特有の履歴やそのデータと用いられかたに関する機密維持のための情報を保持する、あるいは多数のブロックを束ねたカプセルとして扱う等の機構を実現することはもはやストラクチャー・マトリクスの延長上の機構では実現できない。

【0028】ストラクチャー・マトリクスではモデルによる演算結果を渡してさらに大きなモデルを形成するための連鎖とその多階層による扱いの考えが無く、これらの扱い、およびその組み替え、切り離しによる運営の考えが、実現できない。現象自身がループを成すもの、分岐を成すものを扱えない。

【0029】これら既存の方法では、広く非金銭価値と

金銭価値を取り込みモデルを能率的に作成する方法、さらに異なる組織間の市場サービス価値を表現するモデルグループ間の数学的対応によるモデル構築、コストドライバーの活動量を基準とした多階層組織への原価配賦モデル構築、部品展開による所要量計算と組み上げコストモデル、経営上の各種の消費源泉、発生源泉およびコスト源泉等を形成モデルの底辺から階層構造にとらわれることなく集約できるモデル（例えば、環境会計が必要になれば、その時点で集約モデルを組み込める）等に対する既存モデルをベースにした方法論が提示されていない。

【0030】さらに、一般にモデルの運用にあたっては、異なる背景を持つ人々の作成したモデルを連鎖したり、切り離したり、組み替えて運用することが要求される。このようなコラボレーションの方策は現在のところ提示されておらず、これらの問題を解決するには、現状の概念と構造の延長では対応不能である。

【0031】

【課題を解決するための手段】本発明は、上述に鑑みて現状におけるBSMアプローチの問題を総合的に解決をすべくなされたものであり、この分野の業務に携わるエンド・ユーザが企業現象や社会および自然の事象の因果関係を、あたかも鏡のように有機的な関係として詳細まで投影し、透明性を備えた可視性（Visibility with Transparency）を容易に保持するモデルとして、誰でも理解しやすいテーブル群上に表現でき、そのコンピュータ上のモデルの表現から、逆にその事象を知り得るというモデルの上で情報を設定し、演算した結果を現象に照らし合わせて解釈し、さらにモデルを変更し、組み替え、近未来の現実的なよりよい条件を探るための仕組みを提供することを特徴としている。（

【図1】現象とモデルの2方向性 参照）

【0032】上述の目的を達成するために、本発明に関わるモデルを作成・運用するモデル情報処理システムは、経営現象や社会および自然の事象の因果関係を構造として把握して抽出し、因果関係をなす全ての情報と論理を人間が接しやすいテーブルに置き換え、それをハウジング群とそれらの繋がりが構成する構造体に反映し、この上で演算し、変更・組み替えを伴う処理を実行し、要因源泉に還元して把握でき、かつ将来への対処案を探ることを狙いとする構成母体を提供できることを特徴とする。

【0033】（構造体のねらい）このためには、対象とする事象を、その要素間の関係を総当たり的にとらえ、分かりやすく表現でき、演算性ある因果関係とデータで構成するモデルとして設計し、その結果を構造体に反映し、この構造体を演算して事象の分析と近未来の方向を探ることをねらいとする。

【0034】（ハウジングの概略）コンピュータ上に、モデリングの中核構造であるハウジングにハウジングを入

10

20

30

40

50

れ子としうる構造体をモデリングの中核の構造として、ハウジングを垂直あるいは水平に連鎖する階層のモデリング構造体を構築し、各ハウジングのレベルでテーブルおよびテーブル操作による“透明性を備えた可視性”を提供し、連鎖およびその内容の変更・組み替えにより、演算性ある多様な因果関係モデルを構築し、これを演算実行し、設定ケースの予測、ケース間差違の分析等の結果を得て、現象の実体により、現象の解明、近未来の方向を探索できる手段を提供する。

【0035】(ハウジングの階層化)ハウジング相互の階層化は、上位のハウジングに下位のハウジングを登録することにより成され、上位のハウジングの駆動時に下位のハウジングにデータを渡し、下位のハウジングに入っ

て演算を実行し、その演算結果を上位のハウジングに取り込みすることにより、より大きなハウジング群として大きなモデル構造体を形成でき、下位のハウジングは複数の上位階層の入れ子になるようにも登録可能とする。

【0036】これにより、ハウジング単体の構造と運用

の知識があれば、階層を越えて、シームレスにモデル構造体を構築し、運用でき、ハウジング単位の分担による

コラボレーションを容易にしており、上位または下位の

ハウジングを柔軟に組み替えることにより、数多くのケーススタディを可能とする。

【0037】このため、構造体は、対象とする事象の規模、取り扱い規模、構築順位や履歴によって、水平方向の因果の繋がり、および垂直方向の因果の繋がり、両方向のニーズに応じて対話方式で対応でき、階層は、必要に応じて、随時応じられるハウジングを基本の単位として

構成することを手段とする。(

【図2】 ハウジングを入れ子とする階層構造体の例 参照)

【0038】(複数の上位ハウジングへの入れ子)特定のハウジングに着目すると、複数の上位のハウジングを親とする入れ子関係の登録を変更し、従属関係を固定した場合を除き、ダイナミックな因果の組み替えにより、多様なケースにも応え得るものとする。

【0039】(基本メインライン演算、メインライン演算)一個のハウジングを貫くモデル演算をメインライン演算といい、上位のハウジングの基本メインライン演算を実行すると、これに入れ子となっているハウジングも一貫するモデル演算の流れの中で受け渡され、その入れ子ハウジングでメインライン演算が実行され、結果情報を受け取り上位の基本メインライン演算に戻る基本メインライン演算をベースに、途中で演算結果によって流れを変える、あるいはループ処理を含めたハウジング単位のメインライン演算を可能とする。

【0040】(基本メインライン演算)基本メインライン演算は、四則演算、マトリクス演算(ベクトル演算を含む)、拡張マトリクス演算、級数演算、マトリクス級数

演算、論理演算、多数決論理演算、しきい値演算、部分的ループ演算、文字配列マッチング、条件計算、そのほか本モデル情報処理システムの組込関数、ユーザ登録の組込関数、などを要素としており、さらにこれらをまとめ単純化した演算、これらを円滑に行うためのチェック機能、テナントのボディ情報へのアクセス機能などからなる。さらに、ここで演算とは、データの入力、設定、加工およびデータの受け渡し、転写あるいはそれに伴う加工、テナントの集約と分解を含み、指定による参照、タイミング合わせによるダミー動作およびこれら演算に付随するデータ管理、演算管理、さらには入れ子のハウジングへの受け渡しの演算などをも含む。

【0041】(ハウジングの構造)ハウジングは長方形に広がる格子棚状のフレームからなり、その一つの区画をルームといい、その中にテナント名を持つテナント情報、あるいはテナントを束ねカプセル・テナント名を持つカプセル・テナントを登録あるいは収納し、さらに、これらの収納情報群と収納に関する処理、他のテナント／カプセル・テナントとの相互演算の関係、情報の移転

の関係を収納する。
【0042】(ハウジングの区域)ハウジングは、上辺縁の上辺部、左辺部の左辺部、両者に囲まれた中央部の3機能区域に属するルームからなり、左辺部から上辺部にテナント／カプセル・テナント情報が、あたかも、原点(3区域の接する点—左上部に存在—を原点という。)を周回して送られると理解し、中央部には上辺部あるいは左辺部で決まる演算とテナント／カプセル・テナント情報を収納する。(

【図3】 ハウジングの区域構成 参照)

【0043】(カプセル・テナントの扱い名称)ハウジングのルームには、テナントおよびカプセル・テナントの情報をそれぞれ特定できる名称を付し、特にカプセル・テナントには、カプセル・テナントであることをエンドユーザにも本モデル処理システムにも一覧して識別できる記号または文字列を名称の一部に付して収納する。(

【図4】カプセル・テナントの例 参照)

【0044】(テナント／カプセル・テナントとのアクセス)ハウジングのルームとそのテナント／カプセル・テナントの配置とその名称、その配置イメージのままルーミング・テーブルに登録し、その要素および随伴するプロパティ情報をアクセスして、テナント／カプセル・テナント情報にたどる、あるいは逆に戻れることとし、テナント／カプセル・テナントへは、ルーミング・テーブルの表示画面経由で、これらテナント情報群にアクセスし、表示し、変更・設定などの操作が可能であり、ハウジングの下位の入れ子へは、ルーミング・テーブルの入れ子として区域毎に登録しているカプセル・テナントをクリックし下位のハウジングにアクセス可能とする方法による。(

【図5】ハウジングのテナントへのアクセス 参照)

【0045】(ハウジングの有効メインライン演算域) 中央部を縦方向にみた場合、ハウジングにおいて、列／行方向に見た場合、上/左辺部のルームにのテナントまたはカプセル・テナントが配置され、かつ中央部のルームに少なくとも一つのテナントまたはカプセル・テナントが配置されている列／行領域を有効メインラインの対象演算域とする。(

【図6】 ハウジングの有効メインライン演算域 参照)

【0046】(中央部のルームに登録の演算) 中央部のルームにテナントのヘッダー情報であるテナント名の先頭部分に配置したエンドユーザに分かりやすく、本モデル情報処理システムも認識しやすい演算記号、あるいは関数とパラメータ、さらに、その演算記号が配置されたテナントに随伴する固有の多次元テーブルの固有名を登録でき、この随伴テーブルは、対応する上辺部の直上のルームのテナントと縦方向演算し、その結果を、その中央部のルームに保存し、さらに縦方向の並列演算の結果があれば横方向の演算を行い、演算結果は左辺部の直左

【0047】(因果関係と演算の対応) エンド・ユーザは、ハウジングの3領域に表現された要素をたどることによりコンピュータがどういう演算ステップを実行しているかを、容易に知ることができ、インプット情報、その演算結果とアウトプットとの因果関係を目視で理解できる仕組みとする。

【0048】(先決、中間結果、後決) 基本メインライン演算はハウジングの有効演算域における演算で、上辺部にあって左辺部にはないテナント／カプセル・テナントを先決テナント／カプセル・テナントといい、両者に存在するテナント／カプセル・テナントを中間テナント／カプセル・テナント、左辺部にのみ存在するテナント／カプセル・テナントを後決テナント／カプセル・テナントといい、これら3者を巡る基本メインライン演算のシーケンスを確立することを特徴とするもので、先決テナント／カプセル・テナントから出発し、上辺部、中央部および左辺部のテナント／カプセル・テナントで決まる演算を経て、中間テナント／カプセル・テナントに結果を得、原点を周回して上辺部のテナント／カプセル・テナントに中間結果を得、全ての後決テナント／カプセル・テナントに結果を得て終わる一連の演算の流れをいい、その流れのなかで入れ子ハウジングが登録されている場合は、それとのデータの受け渡しと、その入れ子のハウジングの一連の演算は下位のハウジングの演算として行われる方法とする。

【0049】(基本メインライン演算のループ演算) 現象がループによる因果関係を成し、これを反映するハウジングが演算による収斂結果を必要としている場合は、そのハウジングを一体とするループ演算を設定し、その

演算を実行できるとする。

【0050】(ハウジングにおける基本メインライン演算以外の演算) 基本メインライン演算、ハウジングのループ演算などハウジングを一巡する演算をメインライン演算といい、他に、これに先立つ、モデルの登録・設定、テナント情報の入力・設定、モデル外部からの入力、基本メインライン演算後の結果の表示・確認、分析、モデル外部への出力など、操作、運用、管理の面での演算、また、基本メインライン演算、メインライン演算などをも含めたナビゲータなどによる自動実行など、メインライン以外にも演算が行えるとする。

【0051】(テナント情報) ハウジングのルームに収納する情報には、エンドユーザが代表名で扱うテナント名、あるいはカプセル・テナント名のもとに、演算に使用する、リスト情報、多次元テーブル、ファイル、データ・ベース、入れ子である下位ハウジングの構成区域名、の他に管理用のプロパティ用の文書情報を含み、特に、中央部のルーム収納情報には、テナント名の一部に特化した演算形態である演算記号群を含み、これに随伴する演算情報群として、リスト情報、多次元テーブル、ファイル、データ・ベース、入れ子である下位ハウジングの中央部区域名、管理用のプロパティ用の文書情報をも収納する。

【0052】(繰込・繰出情報のレベル) テナント／カプセル・テナントの情報は、単にメインラインの演算に関するものだけでなく、入力などの準備段階やメインラインの事前段階(繰込レベル)、実行後の表示・出力段階(繰出)など、レベル分けを行って、状態に応じた管理を行い、複雑な演算、分かりやすく扱いやすい表示、入出力作業などを容易にし、レベル間の変換演算等を明確にし、ルームに収納する。

【0053】(項目リスト) テナントに収納する情報の形式が多次元テーブル(スカラー、ベクトルを含む)の個々の要素に関する意味付けとテーブル内の位置に関し、テーブル軸に随伴する項目リストを参照して、表示、外部との入出力、メインライン演算あるいは保管し、各ハウジング内で共用できる項目リストを軸情報として利用しうる。

【0054】(項目リストの共通化) ハウジングでテナントの使用項目リストを共通化することにより、プログラミング作業を回避し、対話型操作で項目の追加・挿入・削除・名称変更などの作業に先立って影響を受けるテナントをあらかじめ操作者に知らせ、これら変更作業を可能とし、同時にテナント情報の追加・挿入の自動枠取り、自動削除を行う。

【0055】(多次元テーブルの軸対応) ハウジング全体でルームに収納するテナント／カプセル・テナントの多次元テーブル(スカラーおよび一次元を含む)の軸の方向について統一して随伴する項目リストを扱いことを可能とする。(

【図7】ハウジングの区域別の多次元テーブルに関するテナント軸の例 参照)

【0056】(中央部のテナントにおける項目リストの借用)中央部のテナントが多次元テーブルである場合に、随伴している横軸および縦軸の軸情報については、特に明示的に指定しなくても、それぞれ、上辺部にあって直上のテナントの随伴情報である軸1、左辺部にある直左のテナントの随伴情報である軸1の項目リストを借用することで、中央部のルームにテナントを明示的に指定を省略することもできる。

【0057】(多次元テーブルの画面表示)テナントの情報表示においては、左上をスクロール原点として、軸毎の詳細情報である項目リスト項目を多次元テーブルの要素位置に対応させて画面上に表示する。3次元以上のテーブルは、軸とその項目を特定し2次元表示する。(

【図8】ハウジングの区域別の2次元テーブルに関する画面表示の例 参照)

【0058】(演算における異なる項目リストの事前マッチング)テナント間の演算において軸対応が取られても随伴する項目リストが異なる場合には、項目名のマッチングによるテナント要素の配列整合を行って、演算を行い、左辺部への結果の収納に当たっても結果のテナントが随伴する項目リストに合わせて配列整合を行って収納する。

【0059】(不確定項目を含む項目リスト)外部からデータを読み込む際にテナントの要素内容と同時に項目名を読み込み新たな項目名を設定する場合、既存の項目名と読み込んだ項目名がミス・マッチングの場合の追加、あるいはメインライン演算によって結果の項目名を抽出したり、項目名順が変更になる場合を考慮した識別記号を項目リスト名にキャッピングして設定し、エンドユーザおよび本モデル情報処理システムが一覧的な識別を可能とする。

【0060】(他のハウジングに収納のテナント情報の参照)上辺部の先決テナント、あるいは中央部のテナントの情報として、他のテナントに存在するテナントを指定して、セキュリティの制約がない限り、参照可能である。

【0061】(基本メインライン演算のルール)ハウジングの基本メインライン演算は、上辺部に存在し左辺部に存在しない先決テナント/先決カプセル・テナントを出発候補として、上辺部、中央部、左辺部のテナント/カプセル・テナントの組み合わせで決まる縦方向演算および横方向演算を経て、左辺部に得た結果を中間テナント/カプセル・テナントとして、上辺の出発候補に加え、同様な演算を繰り返し、全ての有効メインライン演算域を巡り、後決テナント/後決カプセル・テナントに結果を得るもので、ルーミング・テーブルの設定により演算シーケンスを自動的に確立し、変更にも自動的に応じる。(

【図9】中央部における基本メインライン演算の流れの例 参照)

【0062】(縦方向の協調演算)基本メインライン演算において、中央部ルームにテナント名を登録するタイプによっては、その横方向にこれと協調する従属的縦方向演算を登録し、上辺のテナントの参照とそのテナントが出発候補となるタイミングあわせを可能とする。

【0063】(上・左辺部のテナントと中央部テナントの軸2整合)中央部のルームに収納のテナント情報名を後続文字列で指定することによる2テナント演算においては、中央部のテナントの横軸と上辺部の直上テナントの第1軸(軸1ともいう)を共通軸とし、かつ、縦軸と左辺部直左テナントの第1軸(軸1ともいう)を共通軸とする。(

【図10】縦積・横和演算の場合の多次元テーブルに関する軸整合の例 参照)

【0064】(メインライン演算シーケンスの制御演算)ハウジングの演算シーケンスの流れを制御する演算記号、あるいは関数により、進行する演算結果に応じて、分岐、挿入、インターラプトを可能にする。

【0065】(演算の複数グループの情報出力)演算結果が複数グループの情報となる場合は、左辺部のルームにはカプセル・テナントを指定して出力する方法を可能とする。

【0066】(モデル領域の一括コピーによる新モデル領域の生成)ハウジングにおいて、中央部の特定区域を指定し、これと協調する上辺および左辺の区域と共に複写し、操作者が既存のテナントと同綴名を避ける設定を行って、中央部の新たな始点を原点として挿入する。ここで指定した領域を統合した名称で扱い、表示する方法を可能とする(

【図11】ルーミング・テーブルにおける特定モデル領域の複写の例 参照)

【0067】(モデル領域の一括コピーとその転置モデル領域の生成)ハウジングにおいて、中央部の特定区域を指定し、これと協調する上辺および左辺の区域と共に複写し、操作者が既存のテナントと同綴名を避け、転置識別文字の設定を行って複写し、中央部の新たな始点を原点として挿入する。ここで指定した領域を統合した名称で扱い、表示する方法を可能とする。(

【図12】ルーミング・テーブルにおける特定モデル領域の複写・転置の例 参照)

【0068】(ミニ・ハウジング演算)演算種別の形態の一つとして、ハウジングの上・左辺部の各テナントがスカラー要素で構成される集合領域について、これを上辺部、左辺部、中央部の各一個のテナントで演算できる形態として演算をカプセル化し、ルーミング・テーブル上の占有域を大幅に縮小して一覧時の可視性を向上し、設定に伴う命名作業、項目リストの設定などの作業を省き、演算実行に要する時間を短縮する。

【0069】（ハウジングの多様な連鎖方法）対象に合わせたモデルの拡大と再編成、背景の異なる人数によるモデリング作業のために、モデルの成長、切り分け、統合、操作性の観点から、上位ハウジングの直下において連鎖関係を結ぶハウジングを連鎖メンバーと呼び、下記の多様な形態の連鎖の生成と、演算を可能とする。メンバーが単独で上位ハウジングと入れ子関係にあって、上位ハウジングのメインライン演算のもとで、上位の情報を渡し、メンバーのメインライン演算を実行し、その結果を上位ハウジングに渡す垂直連鎖。上位のハウジングの傘下で、その複数メンバー同志の間で情報を渡し、情報を受け取ったメンバーのメインライン演算を実行し、その結果を他のメンバーに渡しすことを繰り返す、全てのメンバーを連鎖する水平連鎖。前2者の複合として、上位のハウジングとの間で情報の受け渡しを行う複合連鎖。

【0070】（垂直連鎖）垂直連鎖は、下位ハウジングの3区域を、それぞれカプセル・テナントとし、上位ハウジングの対応する区域に登録して入れ子とし、さらに、上位ハウジングの左辺部に下位ハウジングの上辺部をカプセル・テナントとして登録し、上位ハウジングのテナントから情報を受け取る受け渡し関係を設定し、かつ、上位ハウジングの上辺部に下位ハウジングの左辺部をカプセル・テナントとして登録し、上位ハウジングのテナントへの情報の受け渡し関係を設定し、上位ハウジングの演算実行時に、これらの関係を基本メインライン演算の流れに則り垂直連鎖を実行する。

【0071】（異なる階層間の情報の受け渡し）異なる階層間の情報を受け渡すことを、次の方法を整え可能とする。上位ハウジングの上辺部および左辺部に対となる授受のテナントおよびカプセル・テナントを配置し、その交点である中央部のルームにエン・カプセル演算またはデ・カプセル演算を配置して、双方でのテナント名の変換を指定し、カプセル単位の授受を行う。（テナント／カプセル・テナント単位の受け渡し）上位ハウジングの上辺部および左辺部に対となる受け渡しと受け取りのテナントおよびカプセル・テナントを配置し、その交点である中央部のルームに受け渡し連鎖演算を配置して、テナント間の連鎖関係を登録し、テナント間連鎖テーブルに入り、随伴する項目リストの項目間での一致カ所、あるいは、並び順対応等の方法で、テナントの要素毎の授受を行う。（要素単位の受け渡し）

【0072】（垂直連鎖の自動生成）上位ハウジングに下位ハウジングを最小の操作で垂直連鎖させたい場合、操作者が、上位ハウジングで垂直連鎖し挿入格子点を選び、ハウジング・リストで下位ハウジングを選択し、垂直連鎖を上位ハウジングのルーミング・リスト上に自動的に展開する。

【0073】（水平連鎖）上位ハウジングの傘下のメンバーがその相互間で方向性連鎖（ループを形成しな

い）をなす場合、それらメンバー個々の、上辺部、左辺部、中央部の区域を識別した情報を付しカプセル化し、それぞれカプセル・テナントとして、上位ハウジングの上辺部、左辺部については、対応するカプセル・テナントを、中央部については、右下がりの対角線上の対応するルームに配置し、下位ハウジングとして演算実行できるようにし、左辺部カプセル・テナントを上位ハウジングの上辺部に、下位ハウジングの上辺部を上位ハウジングの左辺部に配置して、下位ハウジング間連鎖演算域を形成し、その交差域のルームに上辺の下位ハウジングから左辺の下位ハウジング間の受け渡し関係が存在するカ所に、操作者がハウジング間連鎖演算テナントを登録し、有効メインライン演算域について、水平連鎖が上位ハウジングの演算実行で進行する構造とする。

【0074】（水平連鎖の自動生成）上位ハウジングに下位ハウジングを最小の操作で水平連鎖させるため、操作者が、上位ハウジングで挿入格子点を選び、ハウジング・リストでメンバーとなる下位ハウジングを選択し、水平連鎖を上位ハウジングのルーミング・リスト上に自動的に展開し、有効メインライン演算域を選定し、上位ハウジングとの受け渡しをする下位ハウジングのカプセル・テナントを自動登録する。

【0075】（ローカル領域とグローバル領域）特定のハウジングを越えて、共用あるいは借用するために、他のハウジングの入れ子となるハウジング、テナントあるいは項目リストなどの情報があり、グローバル領域（これに対して、個々のハウジングの領域をローカル領域という）の領域を設定し、それぞれをグローバル・ハウジング、グローバル・テナント、グローバル項目リストと称し、この領域で保管する情報に関する変更は、この領域のみで可能とし、ローカル領域のハウジングに組み込んで使用可能である。（

【図13】ローカル領域とグローバル領域の説明例 参照）

【0076】（ハウジングのオカレンス構造）エンドユーザにとって、理解しやすく取り扱いの利便のため、ほぼ同じ構成のハウジング全体が層を成して重なった構造で、ルーミング・テーブルを共有できるハウジング群をオカレンス構造として扱い、個々の層は単一のハウジングとして扱い、上位ハウジングに登録する。（

【図14】時系列オカレンスの例 参照）時系列現象を扱うオカレンスにおいては、そのハウジング個々の演算実行のタイミングをとる次の方法を可能とする。個々のハウジングに外部からの情報入力終了された段階で、待機段階に入り、全体の上位ハウジングの流れ、演算の処理の流れに沿って演算される。個々のハウジングは待機段階にあっても、上位のハウジングの設定したオカレンス番号に同期をとって演算する。オカレンス関係にあるハウジングの演算として異なるハウジングの指定したテナント情報を上位のテナントを介することなく参照で

10

20

30

40

50

きる。

【0077】

【実施例】本発明に関わる本モデル情報処理システムはBSMの考えを具現すべくモデル構造体のハウジングのエンドユーザによるモデル開発・モデル運用・そのメンテナンスをねらいとして対話方式による画面操作を中心に展開しており、現象分析によるモデル開発、開発モデルの定常的運用のナビゲータ、ネットワークによる多地点のコラボレータによる統合モデルの開発・運用にいたる物である。そのため、ハウジングによるモデルの詳細作成、その実行可能性のチェック、基本メインライン演算、結果の確認を中心に、ハウジングの連鎖、開発された活動量モデルから単位原価モデルへの変換・運用等を中心に展開している。

【0078】（ハウジングを中核単位とするモデル構造体）ハウジングは一般に縦横の柵枠で仕切られるルームにテナントが配置されているというイメージでユーザは把握し、それらが入れ子のハウジングによる階層を成す広がりを持ち、現象をモデル化して構造体に反映していると理解する構造とした。

【0079】（ハウジング階層構造の仕組み）ハウジングは、上辺部、左辺部、両者に囲まれた中央部の3機能区域に属するルームからなり、上辺部および左辺部の各区域内で、テナント名は同綴名(homonym)の存在を許さないが、同綴名のテナント名を両区域が共有し、従って同じ情報を共有することにより、左辺部から上辺部にテナント情報が、あたかも、原点(3区域の接する点-左上部に存在-を原点という。)を周回して送られると理解し、因果関係の演算が進行する仕組みとし、中央部には、テナント名の一部を既に準備されたメインラインの演算群から選択するための指定文字列(記号を含む)として扱い、メインライン演算のフェーズで、演算処理機能を引き出しと演算実行を行う。中央部においては、同綴名のテナントの重複配置を許し、同一の演算形態、あるいは同一の情報を用いることを可能とした。(

【図15】ハウジングの区域と命名のルール(参照)これらハウジングを一貫した演算の流れを基本メインライン演算とし、ハウジング全体として流を計算した結果によって制御し演算する流れをメインライン演算とループ、分岐などを含めた。ハウジングを扱う構造としては、その大きさに左右されずに、その全体イメージを効率よく保持してテナントに細部にアクセスし、演算性を保ち、かつ変更に耐える構造、対話方式でハウジングを扱うために、ルームに収納のテナント/カプセル・テナントに関する情報をヘッダ部分とボディ部分に分けて扱い、それらの名称はルーミング・テーブルに登録し、演算シーケンスの確立、テナント/カプセル・テナントへのアクセスは、このテーブルからアクセスできる構成とした。(

【図16】ハウジング内の名称のつながりの例(参照)

【0080】モデルの表現であるハウジングのルームの配置から、対応する基本メインライン演算を読み出しすことを可能とした。

【図17】にマトリクス演算(縦積・横和)例の読み出し例を示す。(

【図17】テナントの配置から演算シーケンスを読み出す例(参照)

【0081】（基本メインライン演算のロジック）基本メインライン演算のシーケンスは、テナントと入れ子であるハウジングをカプセル・テナントとして一体化したネットワークととらえ確立した。

【0082】（基本メインライン演算のループ演算）現象がループによる因果関係を成し、これを反映するハウジングが演算による収斂結果を必要としている場合は、そのハウジングを一体とするループ演算を設定できる。その方法例は、メインライン演算の出発候補である先決テナント/カプセル・テナントを初期情報として、別時点での再演算のために、そのルームに保存し、基本メインライン演算を実行し、左辺部の収斂判定対象のテナントに満足な結果が得られたか判定し、得られた場合、左辺部に得られた指定のテナントの結果を指定の上辺部のテナントに書き込み、基本メインライン演算、判定、ループによる周回演算を繰り返す。左辺部の収斂判定対象のテナントに満足な結果が得られるまで、ループ演算を繰り返す。(

【図18】ハウジングのループ演算の例)

【0083】（繰込・繰出情報のレベル）テナント/カプセル・テナントの情報には、次の相互に関連するレベル、およびその中で状態に応じ、管理し、表示、入出力作業を容易にし、レベル間の変換演算を円滑にする。(

【図19】テナントの繰込・繰出しレベルの例(参照)

【0084】（項目リスト）テナントに収納の情報の形式が多次元テーブルである場合(スカラー、ベクトルを含む)の個々の要素の意味付けとテーブル内の位置を取り扱うために、テーブルの軸に、項目リスト名を代表名とし、非同綴文字列の項目名で構成する項目リストを軸に随伴した。テナント/カプセル・テナントに項目リスト名が付随する場合、表示、外部との入出力、メインライン演算あるいは保管にあたっては、テナント情報と項目リストを照合して使用され、項目リストはそのハウジングのレベル内でのみ共通に利用できる情報としてハウジングに付属し、そのハウジングのレベル内でのみ項目リストを使用するテナントについて、項目の追加・挿入・削除・名称変更が可能で、影響を受けるテナントをあらかじめ操作者に知らせることが可能となり、さらにこの操作を進めて関連する箇所のテナント情報の追加・挿入の自動枠取り、自動削除を可能とした。

【0085】（項目リストの文字位置指定）個々の項目

リストにおいて項目文字列の位置の特定区間の指定を項目の文字列位置と続く文字列数による指定と、エンドユーザにとって分かりやすい項目欄名称によるカラム位置の間接指定によって扱うことができる。(

【図20】項目リストの項目欄名称によるカラム位置の間接指定の例 参照)

【0086】(不確定項目を含む項目リストの扱い)メインライン演算に先だって項目の内容である項目名を決定できない、あるいは外部からの読み込み、あるいは演算の結果、項目が確定する場合があります、モデル確立時には、不確定であることを、エンドユーザも本モデル情報処理システムも認識して扱うため、項目リスト名に特定の記号をキャピングして扱うこととした。例として、次の場合がある。

? : 外部情報を読み込む場合にテナントの要素内容と同時に随伴情報として項目名を読み込む場合

& : 外部情報を読み込む場合にテナントに随伴の既存の項目名と読み込んだ項目名がミス・マッチを起こし、ミスマッチの項目を既存の項目の最後に追加する場合。

! : 項目名を抽出する場合で、最大項目名を随伴項目としたい場合。

: 項目数の数に変更はないが、メインライン演算の実行で、項目の順が変更になるソートの場合

【0087】(グローバル項目リスト)一つのハウジングを越えて、共通的に使用したい、項目リストを本モデル情報処理システム全体で共用するため、一例として共用識別子記号“[”を付してグローバル情報として、使用することができ、その項目内容の変更はグローバル領域で、使用箇所を照合しながら操作者が変更し管理する。

【0088】(演算のタイプの類型化と表現)基本メインライン演算の流れにおける個々のステップの演算の内容は、中央部のルームに配置するテナント名で使用する記号および文字列の開発時の設定による。この記号ある文字列の設定の演算決定のためのルールは、あらかじめ使用頻度が高い演算は単純な文字または短い文字列で、頻度が低いと思われる、あるいは使用分野が限られる演算については、関数を表す複数文字列として、エンドユーザおよび本モデル情報処理システム双方について分かりやすい一覧性の高い表現として類型化しておき、モデルの開発・設定、あるいは変更の段階で、準備された演算のタイプのリストから対話画面で選び、中央部のルーム名のテナント名の上位文字列として、また、その後の随伴固有情報名とパラメータを簡単な規則で識別可能な文字列で設定できる構造とする。今回の本モデル情報処理システムにおける演算タイプの例を図21に示す。(

【図21】演算タイプの例 参照)

【0089】(テナント間演算の指定の形態)中央部ルームに登録のテナント名は選び方によって基本メインラ

イン演算における縦方向演算の演算方法を決め、並列する縦方向演算が存在する場合は、その横方向演算についても協調する規則を定め、テナント名の先頭部文字列が内部の演算関数を結びつける。演算の形態には、上辺部の直上テナントを使用する一つのテナント(monadic tenant、以降、1項演算という)演算、該当する中央部のルームに固有名を与えて収納のテナント情報名を後続文字列で指定し、上辺部直上のテナントとの二つのテナント(dyadic tenant、以降、2項演算という)演算、後続カッコ内の文字列によるパラメータ指定を用いて、上辺部の直上以外のルーム収納テナントを直接、あるいは該中央部ルームと同一行の位置を間接指定し、直上以外の上辺部テナントをも間接指定する演算(多項演算)を対話画面による選択とプロンプトにより設定可能とする。(

【図22】演算タイプの類型例 参照)

【0090】(ミニ・ハウジング演算の仕組み 参照)上・左辺部が用意された段階で、ルーミング・テーブルの中央部のミニ・ハウジング演算を組み込みたい位置で、演算タイプ選択のプルダウンを選び、ミニ・ハウジング演算を選ぶと、上辺部に左辺部の随伴項目リストが追加されたテーブルが展開し、上辺テナントの項目と左辺項目との因果関係登録のミニ・ハウジング用演算タイプ・リスト(図21 演算タイプの例 に類似)がプロンプト表示され、あたかも、ハウジングのルーミング・テーブルを作成すると同じ操作で入力し、チェックを選択すると、有効演算域について計算可能性のチェックを行い、この演算ロジックが中央部テナントとして保存される。メインライン演算においては、先決値として与えられる上辺部テナントの値が出発候補となって、演算が進行し左辺部に結果を得る仕組みとした。(

【図23】ミニ・ハウジングの仕組み 参照)

【0091】(領域の転置・複写)ハウジング上のある領域の転置・複写をしたい場合、ルーミング・テーブルを表示し、プルダウン・メニューで転置・複写を選択し、領域を指定し(同時に領域名を与えることも可能)、挿入点を指定すると、その領域が転置し挿入され、転置演算記号“T”のパラメータ部(カッコ内)に指定域のテナント名を挿入したテナント名を配置し、異綴にするための文字または記号指定のプロンプトに対応して、指定操作で(図24)では単位原価モデルの意味で、一例として“@”を指定すると、指定領域に協調する上・左辺部域を交互にしそれらテナントの頭部に“@”を付して転置領域に協調する上・左辺部に配置する。これにより、メインライン演算のフェーズに複写指定元の領域に存在する中央部のテナントを共用して、新たな区域に対応する演算処理を行うもので、この仕組みにより、指定領域の中央部テナントが保持する全要素を展開し、それを転置するという作業を省き、中央部の個々のテナントが必要とされる時点で、参照し、転置して

メインラインの演算に使用でき、能率良く処理可能となる。(

【図24】指定モデル域の転置モデルの例 参照)

【0092】(垂直連鎖)垂直連鎖においては、当然ながら、上位ハウジングの中央部に下位ハウジングを区域で分けることなく一括したままで、登録し、上位ハウジングのテナントとの間で受け渡しを定義する方式も考えられるが、一覧性を維持と、演算論理の同一性の観点から、以下の方式を追求した。

下位ハウジングの3部分：上辺部、左辺部および中央部に分け、上位ハウジングの対応する3つの部位に、それぞれカプセル・テナントとして識別できる記号(実施例では、{ }で囲むが、右かっこ省略可能)と、部分を示すキャッピング文字(実施例では、U, L, C)を付して入れ子として登録し、さらに、上位ハウジングの左辺部に下位ハウジングの上辺部を入れ子のカプセルとして登録し、上位ハウジングのテナントから下位ハウジングの上辺部が情報を受け取る受け渡し関係を交互に設定し、かつ上位ハウジングの上辺部に下位ハウジングの左辺部をカプセル・テナントとして登録し、上位ハウジングのテナントへの情報の受け渡しを設定し、上位ハウジングの演算実行時に、下位ハウジングに入って演算実行し、上位ハウジングに戻ってくることにより、上位ハウジングの基本メインライン演算の流れに則り垂直連鎖を実行する。異なる階層間での受け渡しには、モデリング上の切り渡しの都合によって、次の2方法を用意しエンドユーザの便に供するようにした。

【0093】(垂直連鎖の詳細方法)

テナントのまま受け渡す演算タイプ：エン・カプセルおよびデ・カプセルによる受け渡し方法

(

【図25】垂直連鎖-演算タイプ：エン・カプセルおよびデ・カプセルによる例 参照)

テナントの内容を解き要素間で受け渡す演算タイプ：Chainによる受け渡し方法で、カプセルとカプセル・テナント間の受け渡しテーブルを表示し、その受け渡し箇所

の設定に次いで、テナントの第1軸間の整合がとられ、
— 随伴する項目リスト間で項目マッチングを指定するとマッチした箇所に両テナントの要素情報が受け渡される。

— 随伴する項目リスト間の項目名を縦横要素間での、必要箇所を受け渡すため、左辺項目のリストと、上辺項目のリストのスクロール画面が展開し、左辺項目を選び(実施例の場合)、上辺項目をスクロールして選択すると、要素間の受け渡しを確定する方法。

— 項目リストの項目名が異なっても項目順に従って渡せるだけ渡す方法、の3方法を提供している。(

【図26】垂直連鎖-演算タイプChainによるテナント要素間連鎖の例 参照)

【0094】(垂直連鎖の自動生成)垂直連鎖に当たっ

ては、操作者の作業を最小化するため、上位ハウジングのルーミング・テーブル上で垂直連鎖の挿入点を選び、ハウジング・リストで組み込む下位ハウジングを選択した後、次の操作を自動的に行う。上位ハウジングの挿入点に行と列を挿入し、その上辺部、下辺部およびその交点のルームに、下位ハウジングの名称にカプセル記号(実施例では'{')、と区域識別記号(実施例では 'U'、'L')を付した名称を登録する。挿入行の隣接上側に行を、挿入列の隣接右側に行を挿入し、それぞれ、左辺部には上辺部区域識別記号(実施例では'U')を、上辺部には左辺部区域識別記号(実施例では'L')を付した下位ハウジング名をカプセル化(実施例では'{'を付す)して名称を登録する。階層の異なるハウジング間の受け渡しは新たに生成した上位ハウジングの空欄行および空欄列のルームを選びカプセル・テナント-テナント間受け渡し、あるいは段落番号0093の垂直連鎖の詳細方法による要素間の受け渡し方法によって操作者が登録する。(

【図27】垂直連鎖のルーミング・リストにおける設定の例 参照)

【0095】(水平連鎖)水平連鎖においては、上位のハウジングの傘下に下位ハウジングが方向性連鎖をなす場合、その対象となる下位ハウジングを選択し、それらの上辺部、左辺部、中央部の部位を識別できる記号とカプセル記号を下位ハウジング名に付し、上辺部、左辺部については、それぞれカプセル・テナントとして、上位ハウジング上辺部、左辺部に登録し、中央部については、中央部を識別できる右下がりの対角線上の対応するルームに、配置し、下位ハウジングが演算実行できるようにし、さらに、左辺部を上位ハウジングの上辺部に、下位ハウジングの上辺部を上位ハウジングの左辺部に配置して、下位ハウジング間連鎖演算域を形成し、その交差域のルームに上辺の下位ハウジングから左辺の下位ハウジング間の受け渡し関係が存在する箇所に、操作者がハウジング間連鎖演算テナントを登録する。テナント間連鎖については、垂直連鎖で行ったテナント単位で受け渡す、いわば、デ・カプセルとエン・カプセルを組み合わせた方法と、さらに項目間に至ってテナントの要素を受け渡す方法の2種類が可能である。水平連鎖モデルの例として4ハウジング間の連鎖要求を示し、

【図28】には、その連鎖ハウジングの構成成を示し、

【図29】には、水平連鎖ハウジングへの下位ハウジングの登録とハウジング間の連鎖演算域の生成、さらに、テナント要素間の受け渡しに至る仕組みを例示した。(

【図28】水平連鎖モデルの例および

【図29】連鎖ハウジングの生成の例 参照)

【0096】(水平連鎖の演算シーケンス確立)水平連鎖ハウジングにおいて、ハウジング間連鎖演算テナントの登録を終えた段階で、ハウジング間連鎖演算域で、受け渡し関係のない非有効行、または非有効列が存在する

場合は、本モデル情報処理システムが判断し、これらの行または列を圧縮し、メインライン演算のシーケンスの確立を可能とし、したがって、水平連鎖の演算実行順を確立でき、上位ハウジングの演算実行のフェーズを操作者が選ぶことでそのハウジングの演算実行の傘下で連鎖が行われ、下位ハウジングの左辺部に結果を得る。水平連鎖がループ現象を含む場合は、上位ハウジングにおいて、ループ演算を操作者が設定して演算実行する。(

【図29】水平連鎖ハウジングへの下位ハウジングの登録とハウジング間連鎖演算域の生成、および

【図30】連鎖ハウジングの実行順確立と連鎖の仕組み参照)

【0097】(水平連鎖モデルの上位ハウジングへの組込) 水平連鎖モデルが実行可能であると確かめられた段階で、上位ハウジングから情報を取り込んで演算実行するためには、垂直連鎖の方法を組み合わせる。水平連鎖モデルを一個の下位のハウジングとして垂直連鎖として上位ハウジングに組込み受け渡しテナントによって受け渡しを行う。下位ハウジングのルーミング・テーブルを上位ハウジングに複写し、その下位ハウジングと情報の受け渡しテナントを登録して行う。

【0098】(ハウジングのオカレンス構造) ハウジングは、ほぼ同じ構成のハウジング全体が層を成して重なった、いわゆるハウジングのオカレンス構造として扱うことが、理解することが、操作者にとって理解および情報の扱いの面で便利な場合が多く、オカレンス構造の一つ一つの層は単一のハウジングとし扱い、上位ハウジングに水平連鎖モデルとして登録する。時系列現象を扱うオカレンスにおいては、そのハウジング個々の演算実行のタイミングをとる次の方法がある。個々のハウジングに外部からの情報入力終了された段階で、待機段階に入り、全体の上位ハウジングの流れ、演算の処理の流れに沿って演算される。個々のハウジングは待機段階にあっても、上位のハウジングの設定したオカレンス番号に同期をとって演算する。

【0099】(モデル構造体の運営) モデルのカバー範囲が拡大すると、当然コラボレーションが必要となり、個人や責任、専門性の壁、遠隔の壁を越えて運営しなければならないため、

【図31】に示すように、実体情報の在処とエンドユーザーの操作場所は異なり、エンドユーザーに実体感を与えることに応えた。また、モデル開発と一体化した道具として、グラフィック・モデラーの機能、基幹情報処理システムとしての運用として、外部入出力と一体化したバッチ・プロセスを可能とするナビゲーター機能も組み込んでいる。(

【図31】ネットワーク環境におけるモデリングおよびモデルの連鎖と運用 参照)

【0100】

【発明の効果】(事象を容易に把握) 以上、実施例を上

げて説明したように、エンド・ユーザは簡単なハウジングの原理を知るだけで、モデル全体を上位階層で概観し、その詳細を次第に下層のハウジングを辿って概観することにより、コンピュータが処理している因果関係を実体現象と対比して理解することが可能となる。そのため、これまでと異なり、コンピュータの処理内容から逆に企業業務の因果関係、社会事象、自然事象等の因果関係を把握することは極めて容易である。

【0101】(モデルのメンテナンス容易) モデルの処理内容が実体現象に照らして詳細まで、エンドユーザに簡単に理解できるため、これまで、他の方法による巨大モデルはメンテナンス不可能とされていたが、エンドユーザが簡単にメンテナンス可能となった。

【0102】(モデルの移転とコラボレーション) 本発明によるハウジング単位、テナント単位の移転と、容易な多階層化は、モデル構築の分担と組み替えを容易化し、モデルの個人間の流通を可能にしたため、これまで不可能とされた、異なる専門領域に属する人々が開発し運営するモデルを移転し、個別に把握された事象が統合的に把握でき、再開発し、統合し、より大きなモデルに関するコラボレーションが可能となった。

【0103】(トップダウン、ボトムアップ) 本発明のモデル間の受け渡しを表す垂直連鎖と水平連鎖の多様な手法により、既に開発された部分を生かしつつ、トップダウン、ボトムアップ両方向からの要求に迅速に応え、変更がエンドユーザでも可能である。性急な事案の策定や変更にもハウジングを組み替える、プログラミングに依存することなく、対応可能となった。これにより、事業部制からカンパニー制への切り替え事案など、ボトムは替わらないが上位管理方式が変わる事案、製造部門の合併のように詳細なボトム構造が変わる事案など、これまで、詳細なシミュレーションは不可能とされたが、巨大モデルの変更に対しても迅速な対応ができるようになった。

【0104】(未来型) 本発明により、生産・原価などの実績計算を処理するモデルの延長として、近未来の計画システムを実現し、未来の企業行動、社会行動等を探索(Feed-forward Management)ことを可能とし、例えば、管理会計の分野では、過去の分析と言われた分野で、近未来を予見する管理会計を可能にした。

【0105】本来、ループを成す事象をエンド・ユーザが理解すること自体かなり困難を伴い、モデリングに進みにくい状況にある。本発明では、まず、ループを考慮することなく一方向性の因果ネットワークを理解した上でモデルを構築し、その上でループの関係を導入するという段階的アプローチを経ることにより、モデリングを容易に実現する。

【0106】(横方向の演算) 本発明では、モデルの演算記号の役目を縦方向の作用、水平方向の作用とに分けて考えられる。同時に、その因果関係と演算を理解できる

ため、縦方向だけでは解決しがたかった演算を組み込む可能性を広げハウジングの適用領域の拡大の道を切り開いた。例えば、横方向積によって積の連結を能率良く表現でき演算効率を高めた。また、エン・カプセル、デ・カプセル等が可能となり、エンド・ユーザにとって理解上必要な最小単位で進行状況のステップが追い易く、モデルの部分理解のみならず、モデルや演算の変更が能率良く簡単にできるようになった。

【0107】モデルの要素に適した多種類の演算記号を用意することにより多様な経営現象を価値の投入と価値の産出の観点から因果関係を分析し、経営現象の部分モデルを部分から構築して実行し、その結果を確かめながら全体モデルへと対話型により構築が可能であり、構築された全体モデルの部分的な変更、追加、削除などが対話型で可能となる。

【0108】(カプセル化の効果)ハウジングに複数テナントのカプセル化を可能としたことにより、一つの演算テナントが複数のテナントの出力を可能としたばかりでなく、下位ハウジングを3部分に分けてカプセル化し、上位ハウジングに登録することにより、垂直連鎖を容易とし、上位のハウジングだけを意識して、より下位のハウジングの演算が可能となった。

【0109】(エン・カプセル、デ・カプセル)本発明では、複数の情報群をエン・カプセルしたり、エン・カプセルされた複数情報をデ・カプセルして、取り扱いを単純化し、連鎖を簡潔にしたり、テナント単位での情報のモデル間の流通を簡略化した。

【0110】(容易な連鎖の生成)コラボレーションにおいては、多人数によって既に作成され確立されたモデルを連鎖する場合が多いが、上位ハウジングに対し入れ子方式で対処することは、かなり煩雑となるため、垂直連鎖、水平連鎖のプロンプトによる自動生成を準備したことにより、連鎖作業を容易にし、作業性を大幅に向上した。

【0111】(オカレンス)2方法によるオカレンスの処理により、月次処理と期末処理などの複雑な時系列モデル、あるいは、入れ子による階層を扱い、かつ、時系列の同期を取って計算する必要がある階層の深いモデルが処理できることとなった。

【0112】(テナント情報)テナントに多様な種類の情報、例えば、テキスト、リスト、多次元テーブル、ファイル、データ・ベース、イメージ・データ、音声データ、あるいはデータ収納場所情報を登録し、集約して扱うことにより、基本メインライン演算に関する情報以外に、履歴、セキュリティ、コメントなど標準的に扱えることとなった。

【0113】(繰込・繰出)テナント情報に基本メインライン演算に使用する以外に繰込、繰出レベルを設けることによって、エンドユーザの対話操作性、外部との入出力、基本メインライン演算の幅広い能力などを収容する

枠組みを提供した。

【0114】(項目リストとグローバル)テナント情報に項目リスト名を随伴し、項目リストをハウジング共通、あるいはグローバル項目リストとしてBSMシステム共通に扱うことにより、標準化と、項目の変更に対する予知して進めることにより、これまでプログラム段階に遡って変更を要していた自体を回避できるようになった。

【0115】(不確定項目を含む項目リスト)モデル確立時に確立できない項目がある場合に、エンドユーザーも認識しやすく、かつメインライン演算とも連携して処理可能な方式を実現した。

【0116】(項目リストの文字欄指定)項目リスト内の特定文字欄を、エンドユーザが総称名で扱えるようになり、マッチング演算など詳細情報を意識することなく扱えることとなった。

【0117】(ミニ・ハウジング)ハウジングのひとつのルームでハウジングと同等の作用を行えることにより、演算設定を簡単化し、モデルをコンパクトにすることができ、かつ、演算能率をあげることができる。この効果は、経営モデルの財務数値を扱う最終段階で、スカラー値を多く扱う部分をコンパクトにできるモデルで顕著である。

【0118】(モデル領域の一括複写および転置・複写)ハウジング内の特定の機能を果たす領域を一括複写して、モデルの移転を簡略にし、モールド・モデルによる管理を広げることが可能とし、転置・複写により、既にハウジング上で確立した、広い意味の活動量モデルに対応し、単位原価モデルを転置・複写という簡便な操作で実現し、ハウジングの同じドメインで、活動量、単位原価、総額のモデルを確立できるという、企業における根幹システムの実現に新たな道を開いた。

【0119】To Fromリストをハウジングに付属せしめ、その詳細を、テナントの繰込・繰出情報と結びつけ予め登録することにより、自動的に外部情報を必要時点で必要箇所に応じて読み込み、書き出しが可能となり、エンド・ユーザによる一括外部読み込み、あるいは一括演算の実行が可能となり、ナビゲーターによる一環する実行が可能となった。

【0120】本発明により、2000年以上前から曼陀羅に描いていた世界観を、論理計算性を持つ因果関係の世界で実現したことである。

【図面の簡単な説明】

【図1】現象とモデルの2方向性について説明する図である。

【図2】ハウジングを入れ子とする階層構造体の例を示す図である。

【図3】ハウジングの区域構成の例を示す図である。

【図4】カプセル・テナントの例を示す図である。

【図5】ハウジングのテナントへのアクセスの概念を示す図である。

【図6】ハウジングの有効メインライン演算域の例を示す図である。

【図7】ハウジングの区域別の多次元テーブルのテナント軸の例を示す図である。

【図8】ハウジングの区域別の2次元テーブルの画面表示の例を示す図である。

【図9】中央部における基本メインライン演算の流れの例を示す図である。

【図10】縦積・横和演算の場合の多次元テーブルの軸整合の例を示す図である。

【図11】ルーミング・テーブルにおける特定モデル域の複写の例を示す図である。

【図12】ルーミング・テーブルにおける特定モデル域の複写・転置の例を示す図である。

【図13】ローカル領域とグローバル領域の説明を示す例である。

【図14】時系列オカレンスの例を示す図である。

【図15】ハウジングの区域と命名のルール例を示す図である。

【図16】ハウジング内の名称のつながりの例を示す図である。

【図17】テナントの配置から演算シーケンスを読み出す例を示す図である。

【図18】ハウジングのループ演算の例を示す図である。

【図19】テナントの繰込・繰り出しレベルの例を示す図である。

【図20】項目リストの項目欄名称によるカラム位置の間接指定の例を示す図である。

【図21】演算タイプの例を示す図である。

【図22】演算タイプの類型例を示す図である。

【図23】ミニ・ハウジングの仕組みを示す図である。

【図24】指定モデル域の転置モデルの例を示す図である。

【図25】垂直連鎖－演算タイプ：エン・カプセルおよびデ・カプセルによる例を示す図である。

【図26】垂直連鎖－演算タイプChainによるテナント要素間連鎖の例を示す図である。

【図27】垂直連鎖のルーミング・リストにおける設定の例を示す図である。

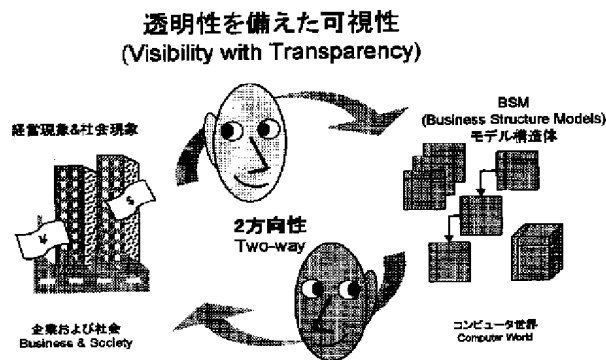
【図28】水平連鎖モデルの例を示す図である。

【図29】連鎖ハウジングの生成の例を示す図である。

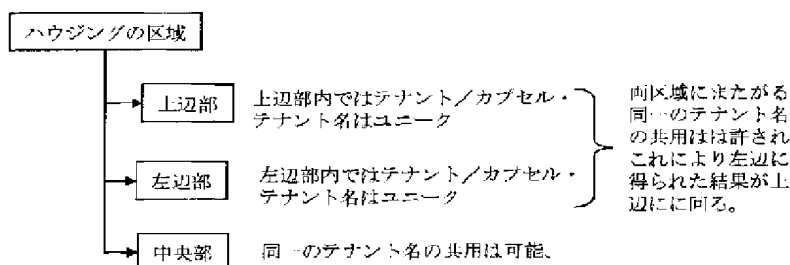
【図30】連鎖ハウジングの実行順確立と連鎖の仕組み例を示す図である。

【図31】ネットワーク環境におけるモデリングおよびモデルの連鎖と運用を示す図である。

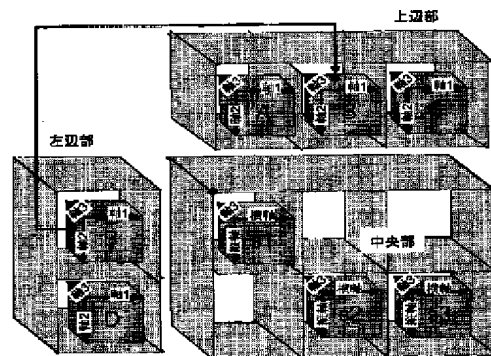
【図1】



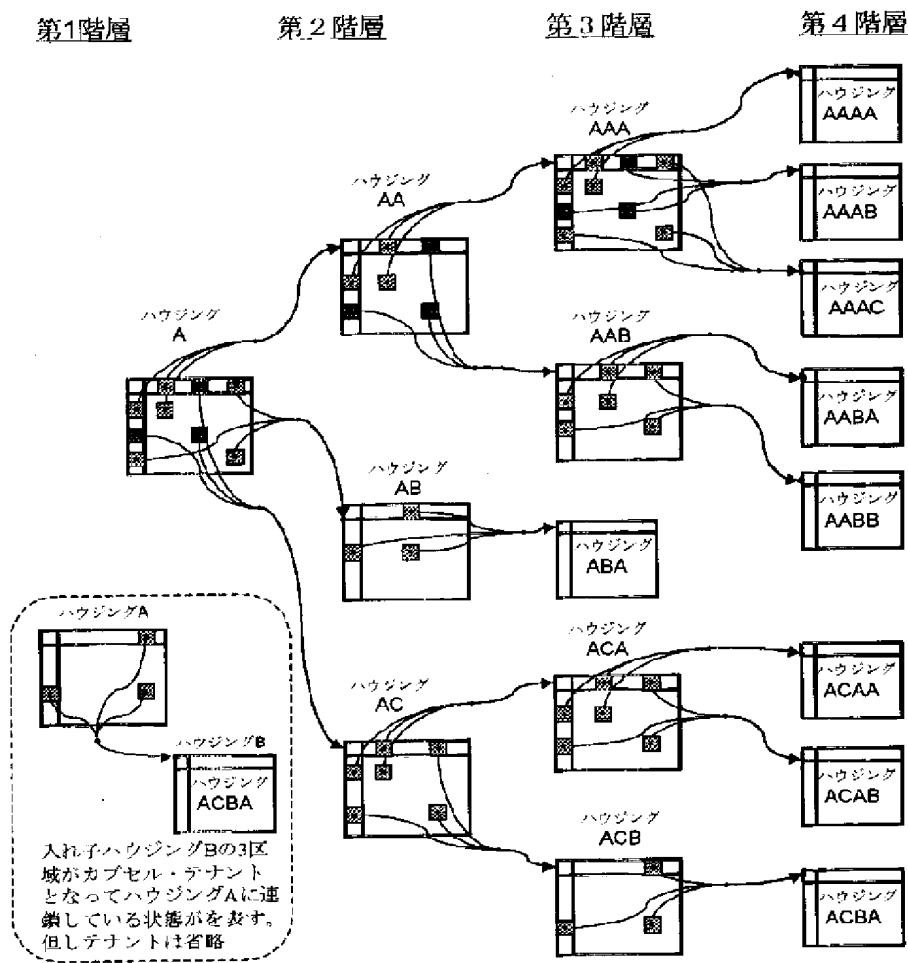
【図15】



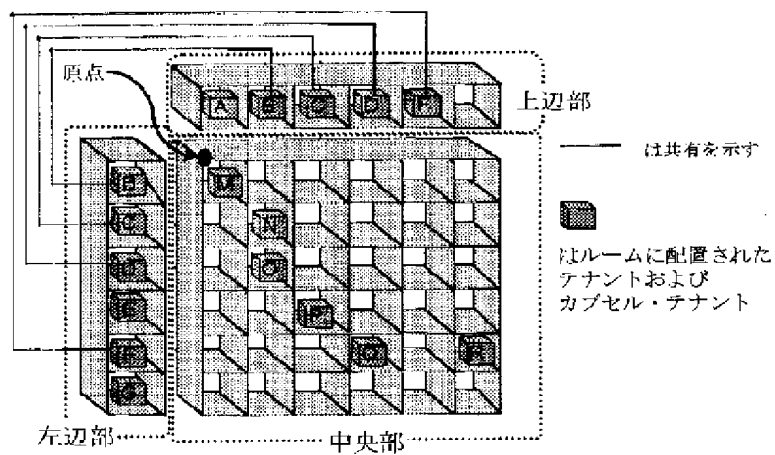
【図7】



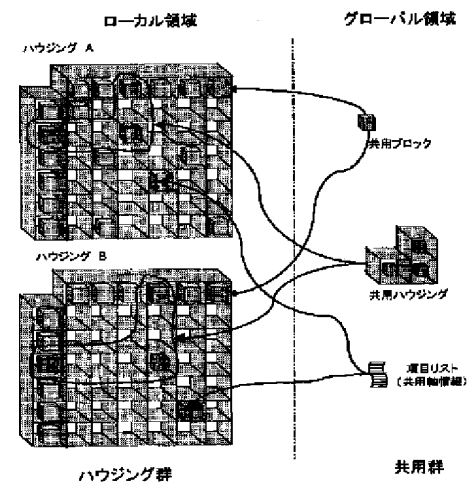
【図2】



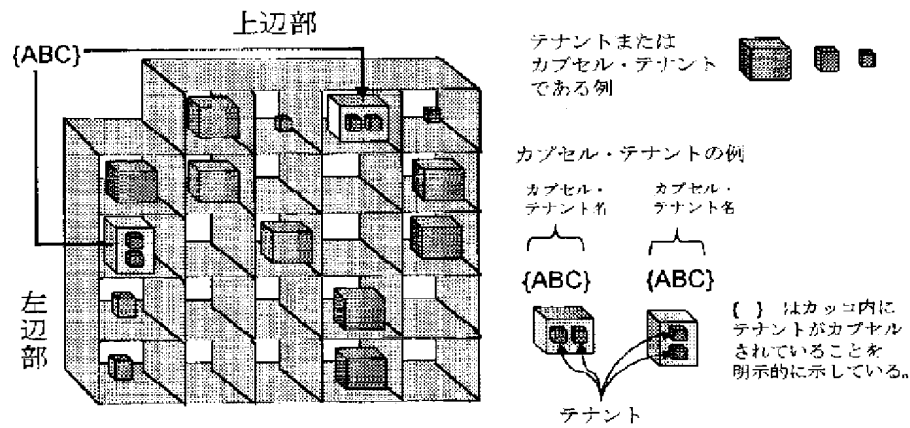
【図3】



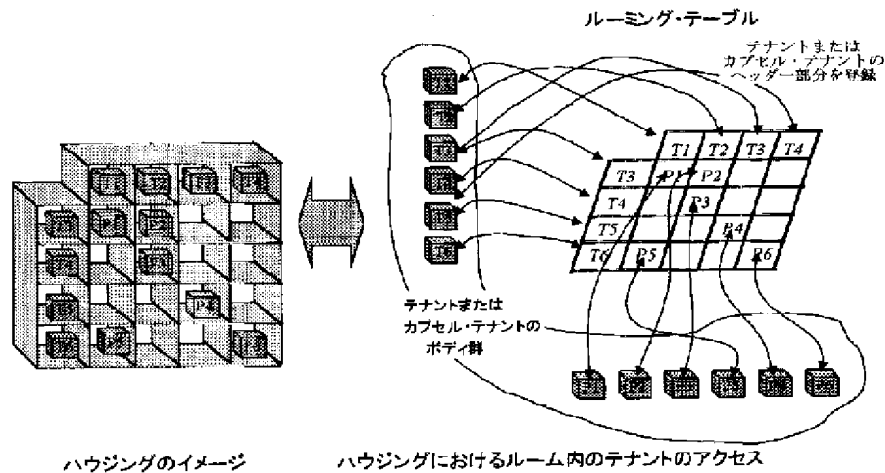
【図13】



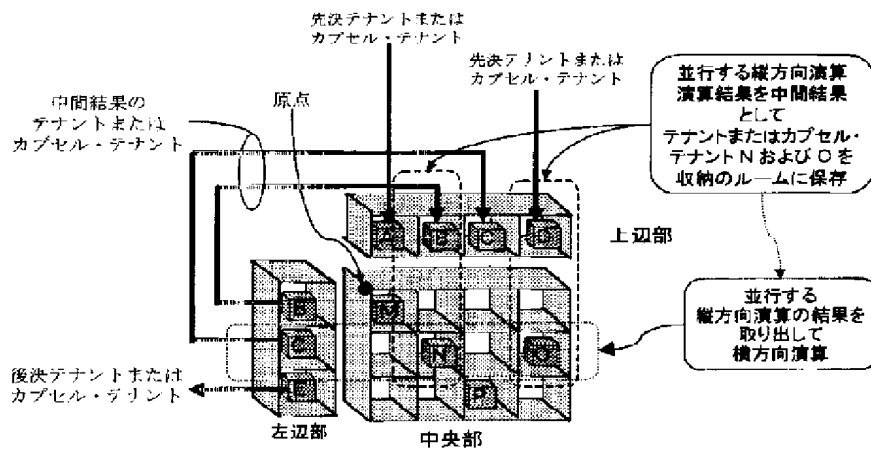
【図4】



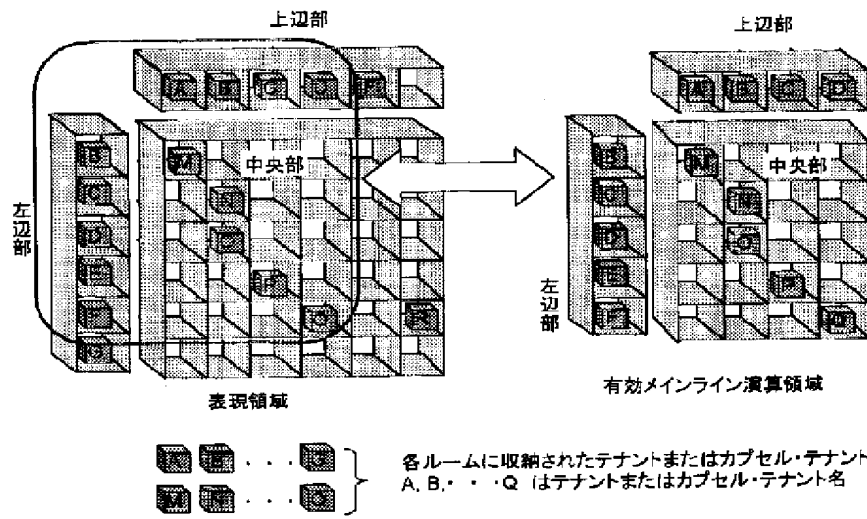
【図5】



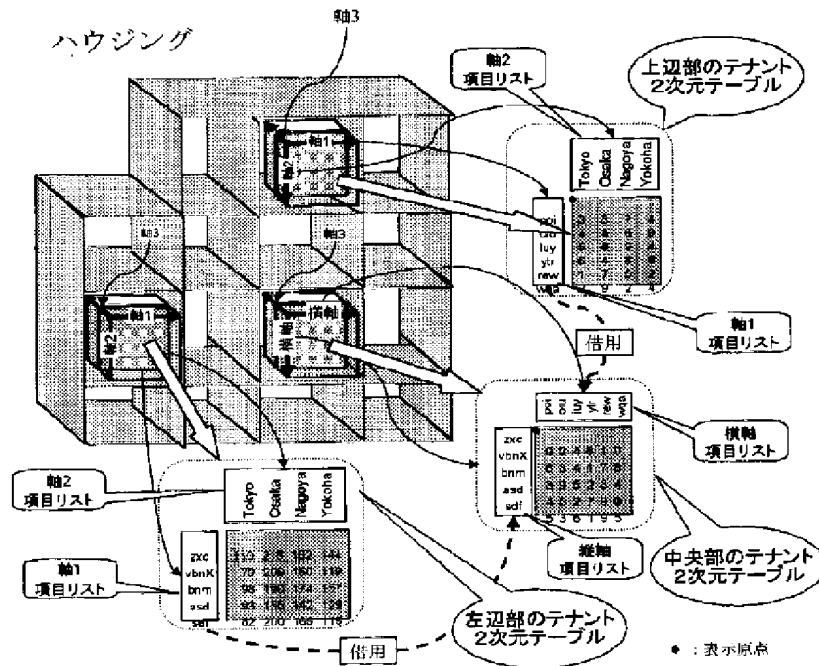
【図9】



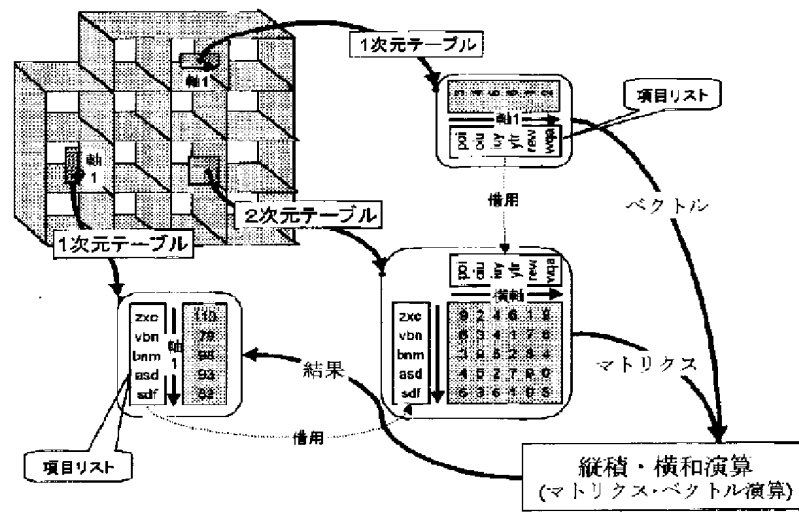
【図6】



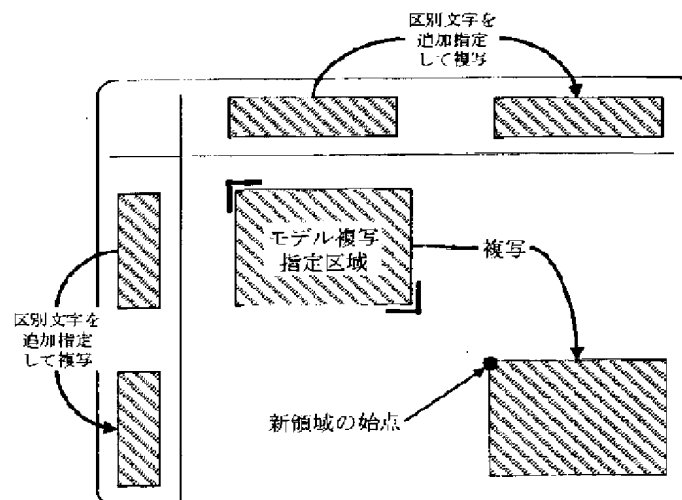
【図8】



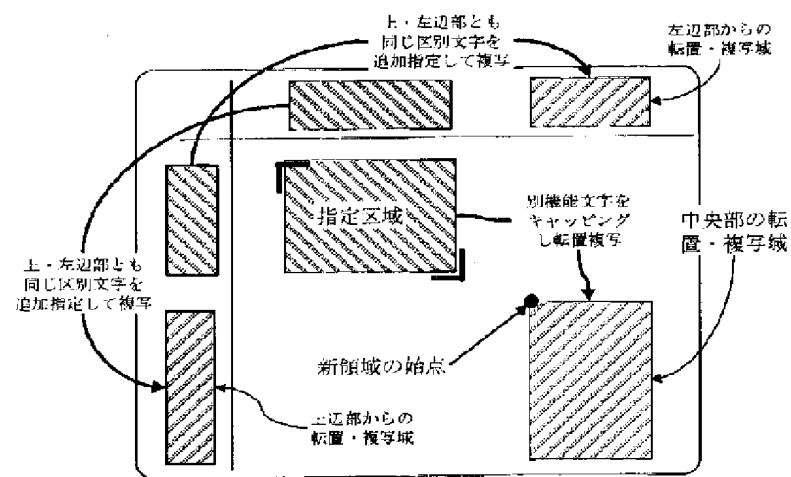
【図10】



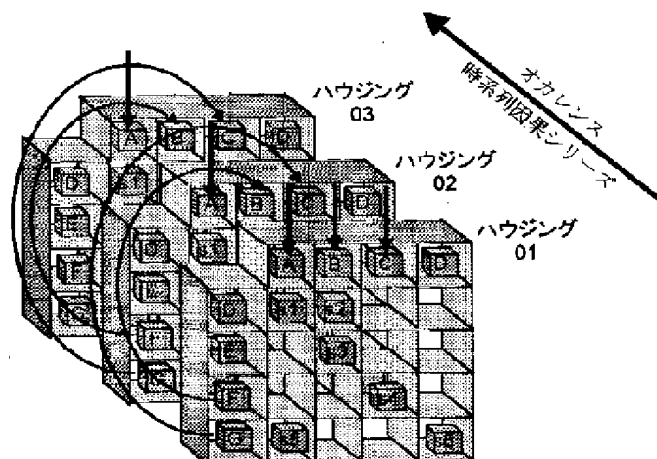
【図11】



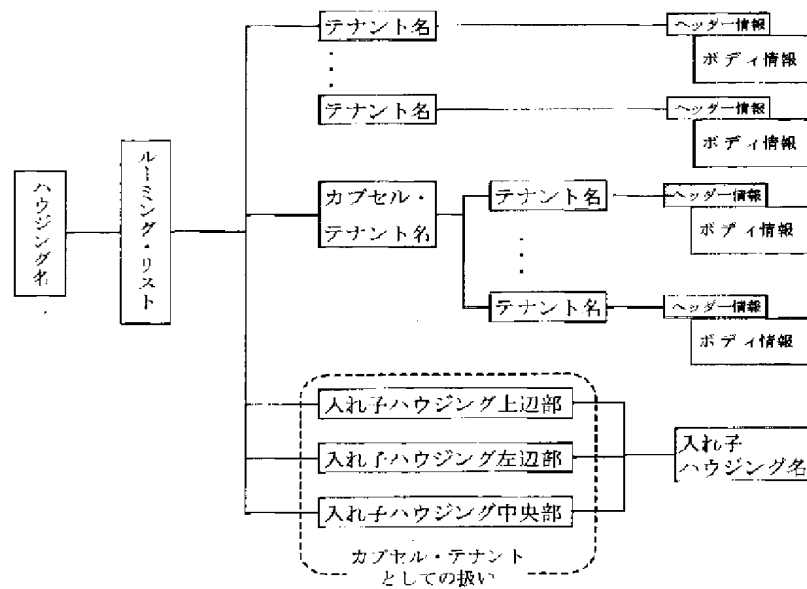
【図12】



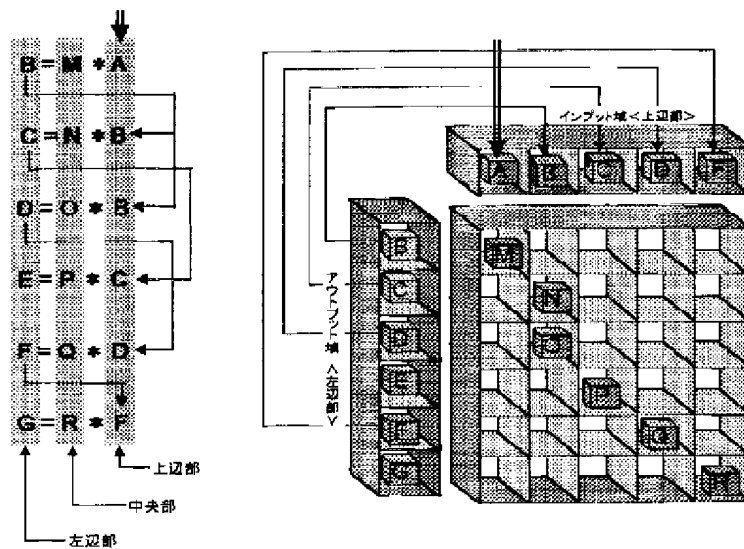
【図14】



【図16】

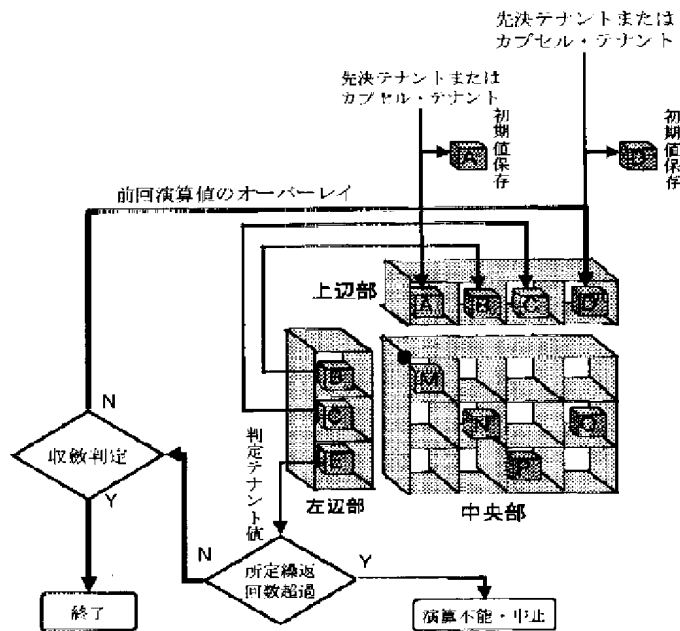


【図17】

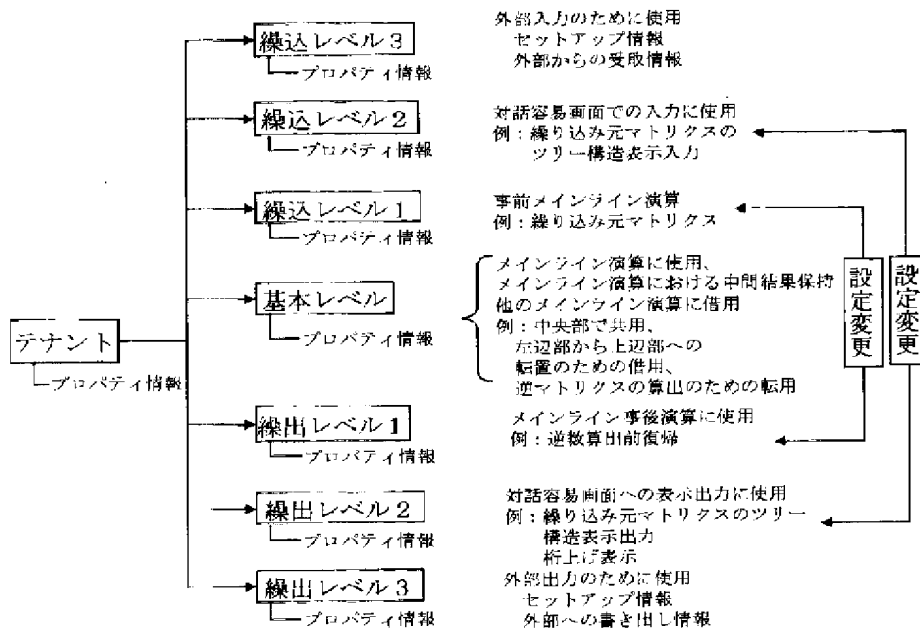


* : マトリックス演算 (一例)

【図18】



【図19】



【図20】

製品販売管理項目のリスト

() 内は文字カラム位置

品目名欄 (1~12)	地域欄 (13~20)	販売形態欄 (21~30)
テレビ	関東	量販店
テレビ	関西	特約店
テレビ	中部	スポット
テレビ	九州	量販店
テレオ	東北	スポット
ステレオ	関東	量販店
ステレオ	関西	特約店
ステレオ	中部	量販店
VCR	関東	特約店
VCR	関西	量販店
DVD	関東	スポット
DVD	関西	量販店

参照カラム位置 → 11111111111122222222223
123456789012345678901234567890

【図21】

演算タイプ	演算タイプの働き	パラメータ部の働き	横方向演算
A()	Add、加算	加算軸指定 倍数指定	横方向加算
b固有な	Bool値と乗積・横和	～	横方向加算
Chain	上辺要素の左辺渡し展開	～	横方向加算
c固有な()	Codition演算、 結果: Bool値	別上辺値の参照	参照タイミング協調 横方向加算
D、 #固有な	グミー演算 上辺の使用可待ち	#指定上辺値の参照	～
Decap()	デ・カプセル	指定名テナント取出し #で順指定の取出し	～
Dup	上辺値を左辺項目に 合わせて複製	～	横方向加算
Xche()	軸上辺変換	2軸のFrom-To指定 バリエーション指定	横方向加算
@固有な	線形ループ演算	～	横方向加算
%()	上辺値の割掛け	別上辺の指定 Rounding指定	参照タイミング協調 横方向加算
*()	上辺値との乗算	別上辺指定	参照タイミング協調 横方向加算
/	上辺値の逆数	～	横方向加算
/()	パラメータ指定 上辺値の割算	別上辺指定	参照タイミング協調 横方向加算
**, **()	べき乗	別上辺指定、 べき乗値指定	横方向加算
*,	横方向乗算	～	参照タイミング協調 横方向 乗算

【図22】

例1 演算タイプに随伴する情報との演算

	AAA	BBB	CCC	DDD
BBB	s1			
EEE		s3		
FFF		sLMN		sMNO
GGG				s4

$$FFF = (LMN \star BBB) + (MNO \star DDD)$$

ここでは \star はマトリクス積を表す。

sLMNは随伴するLMNと上辺とのマトリクス積演算を表す。

sMNOは随伴するMNOと上辺とのマトリクス積演算を表す。

例2 演算タイプにより複数の上辺テナントの演算

	AAA	BBB	CCC	DDD
BBB	s1			
EEE		* (算)		* (算)
FFF			s3	

$$EEE = BBB * DDD$$

*はここでは算術積

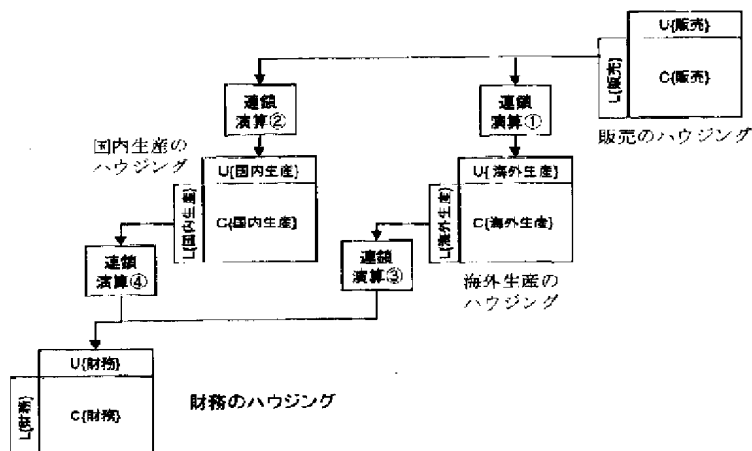
例3 演算タイプによる縦方向演算

	AAA	BBB	CCC	DDD
EEE		sLMN		
FFF	s1	s2		s4
GGG			s3	s4

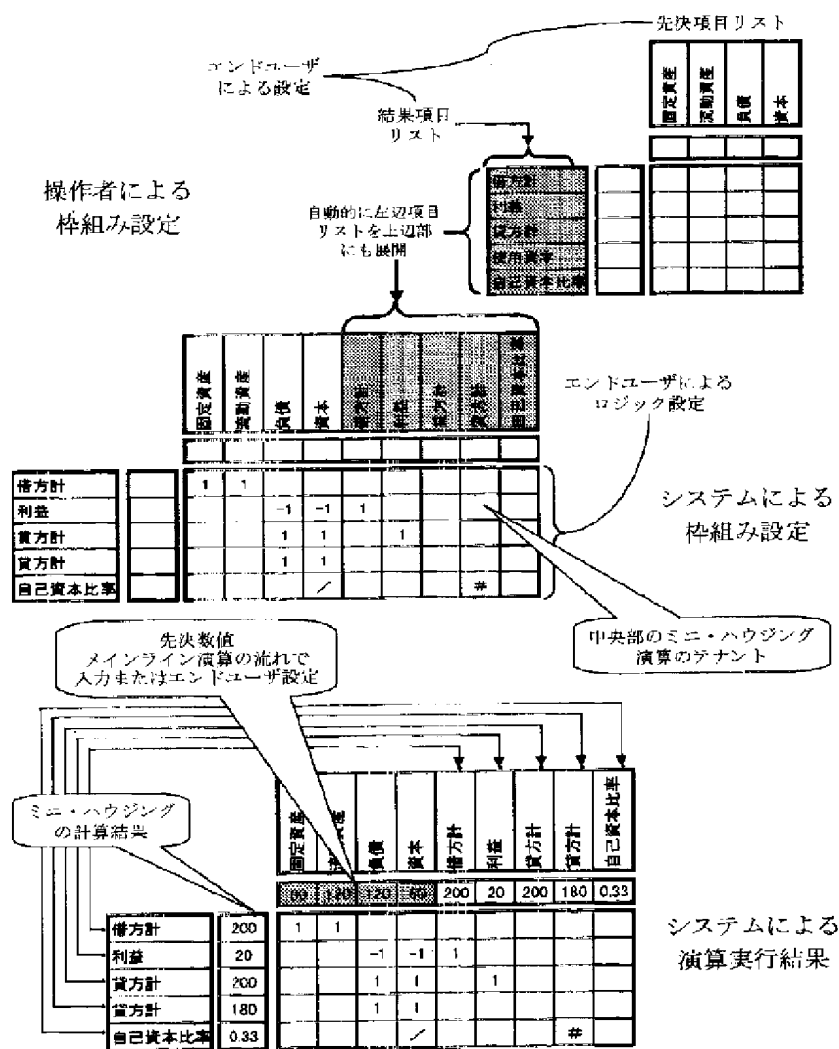
$$FFF = AAA * BBB * DDD$$

*: はここでは横方向演算による算術積

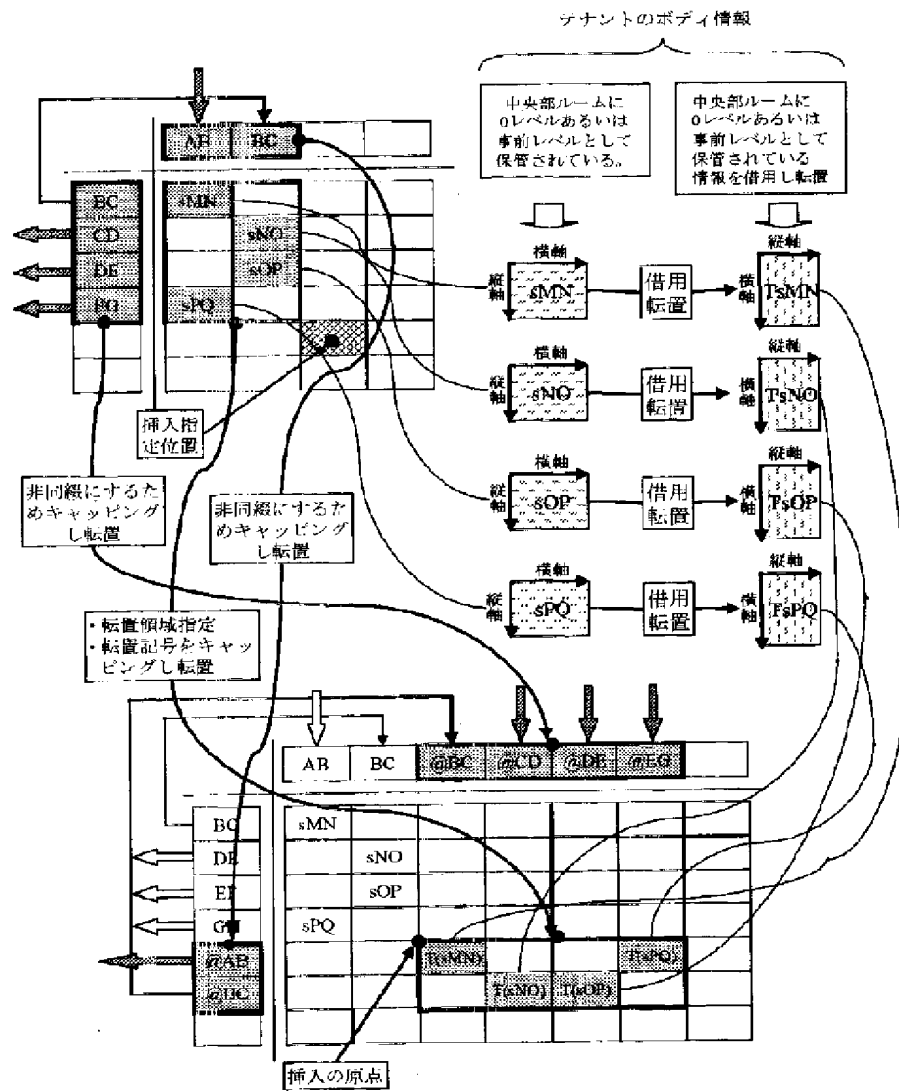
【図28】



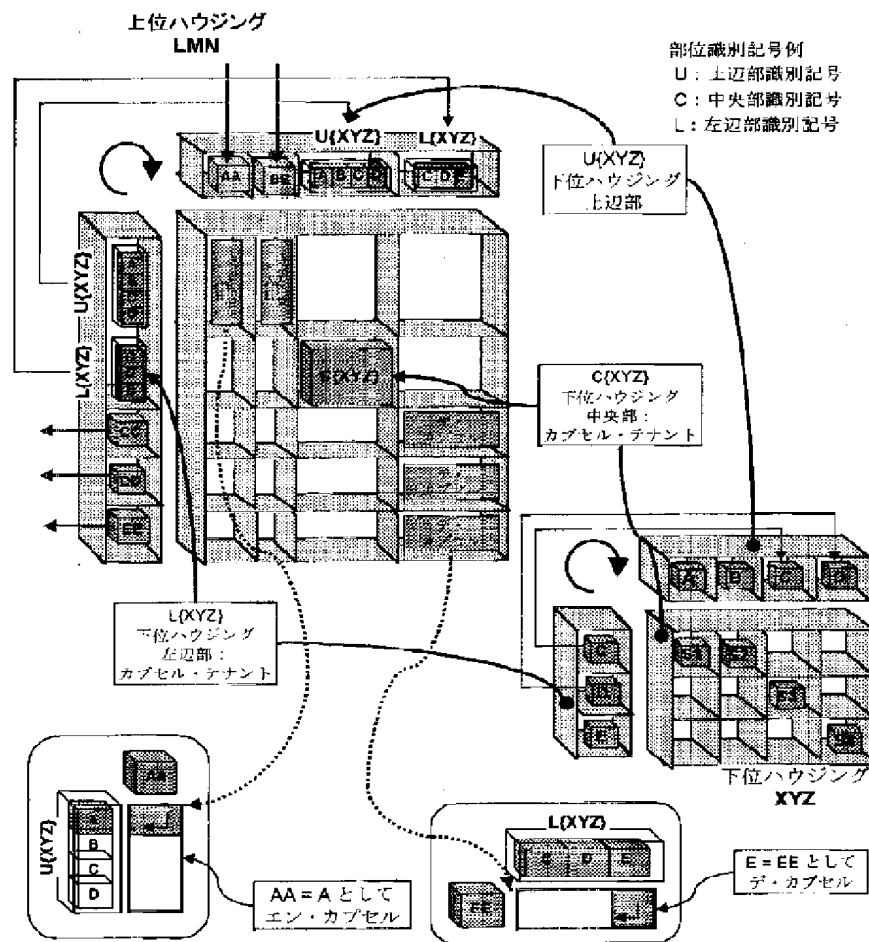
【図23】



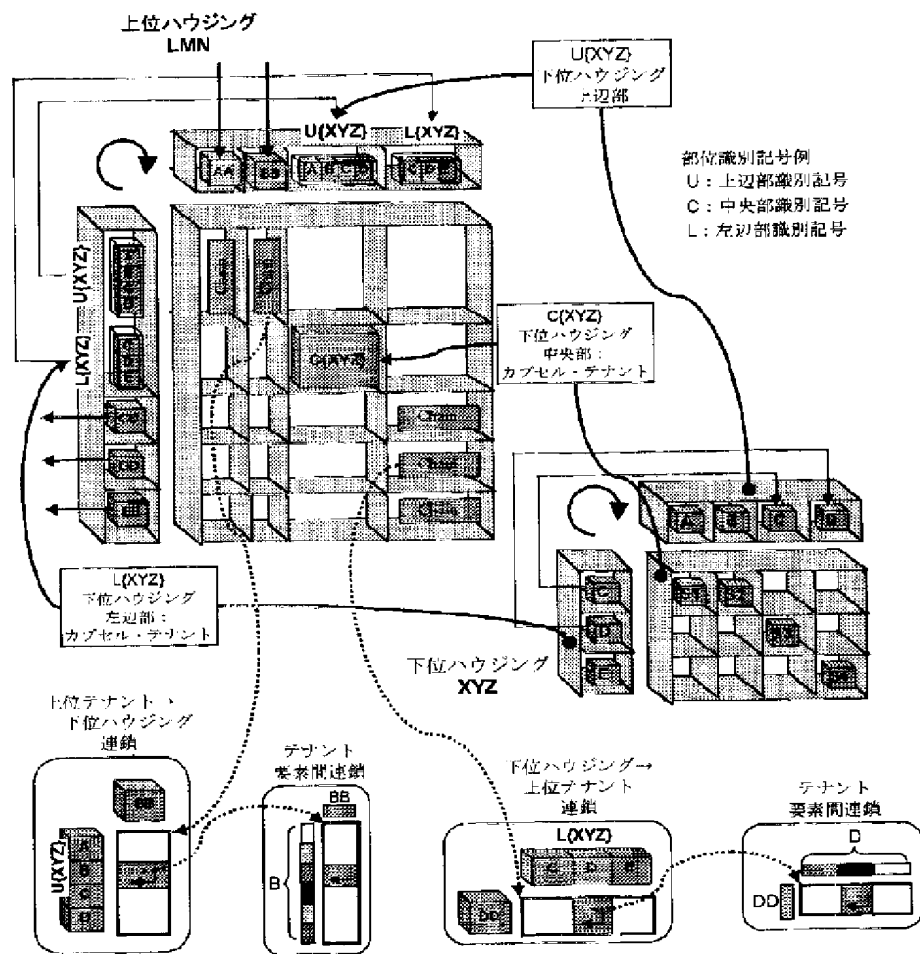
【図24】



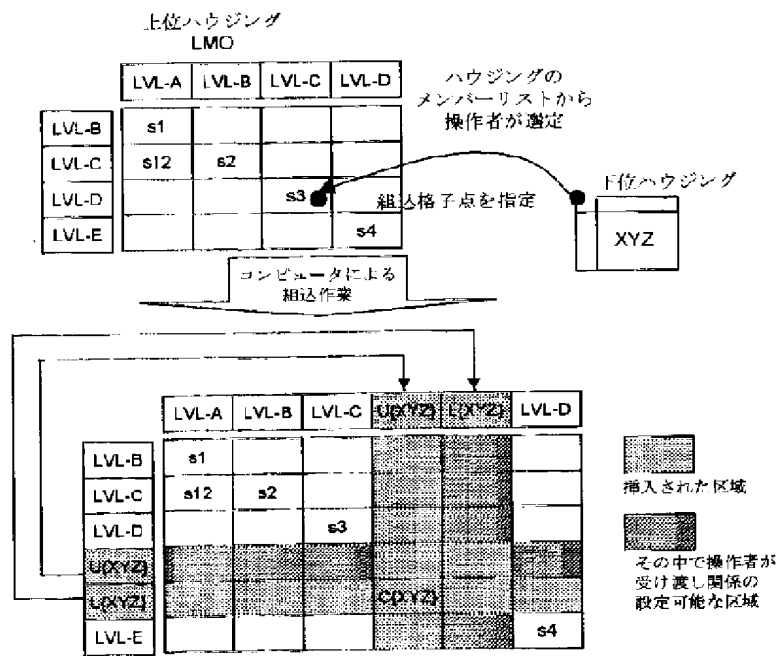
【図25】



【図26】

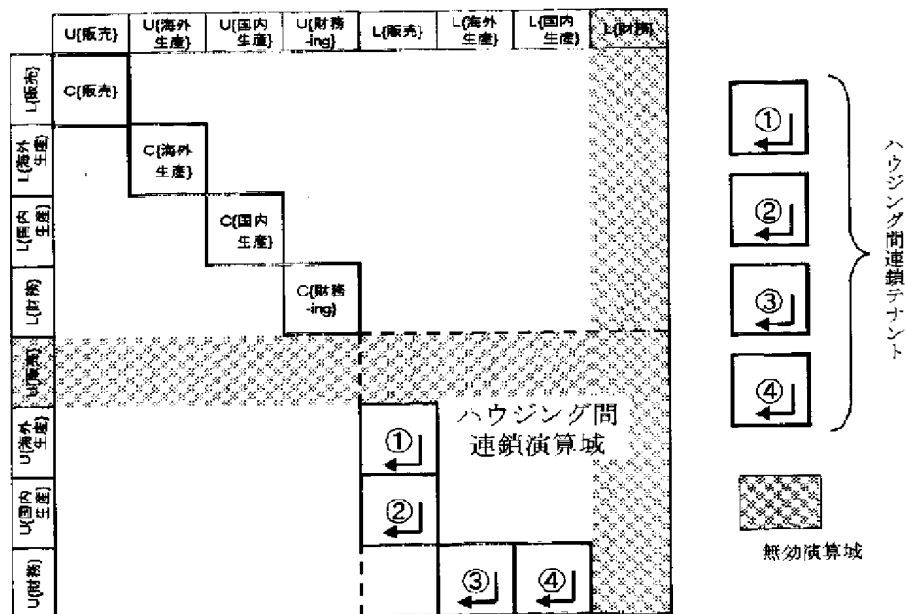


【図27】



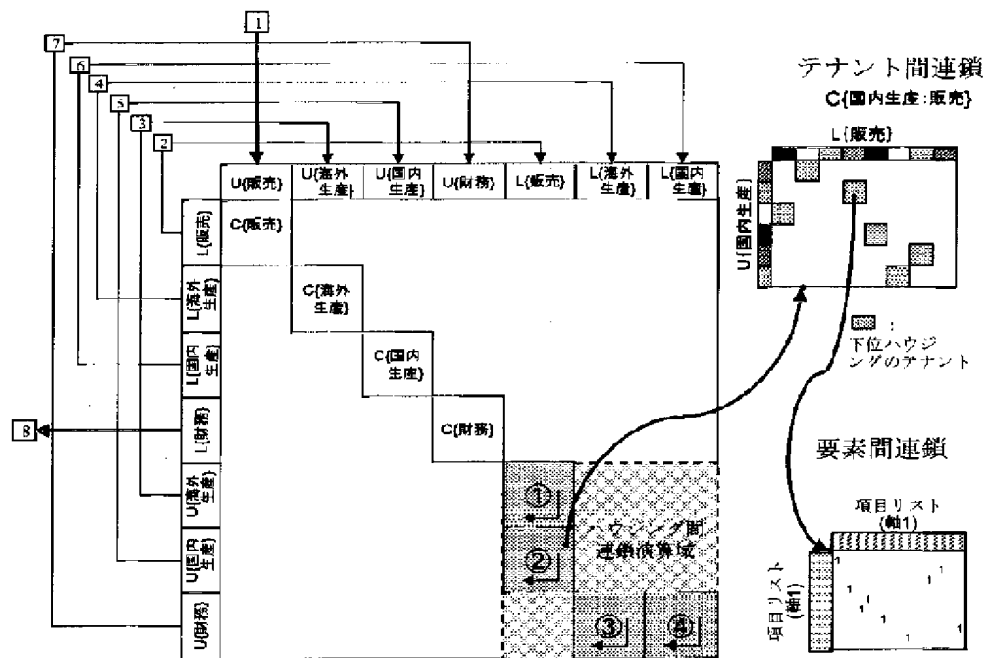
上位ハウジング：LMOに垂直連鎖された下位ハウジング：XYZ

【図29】

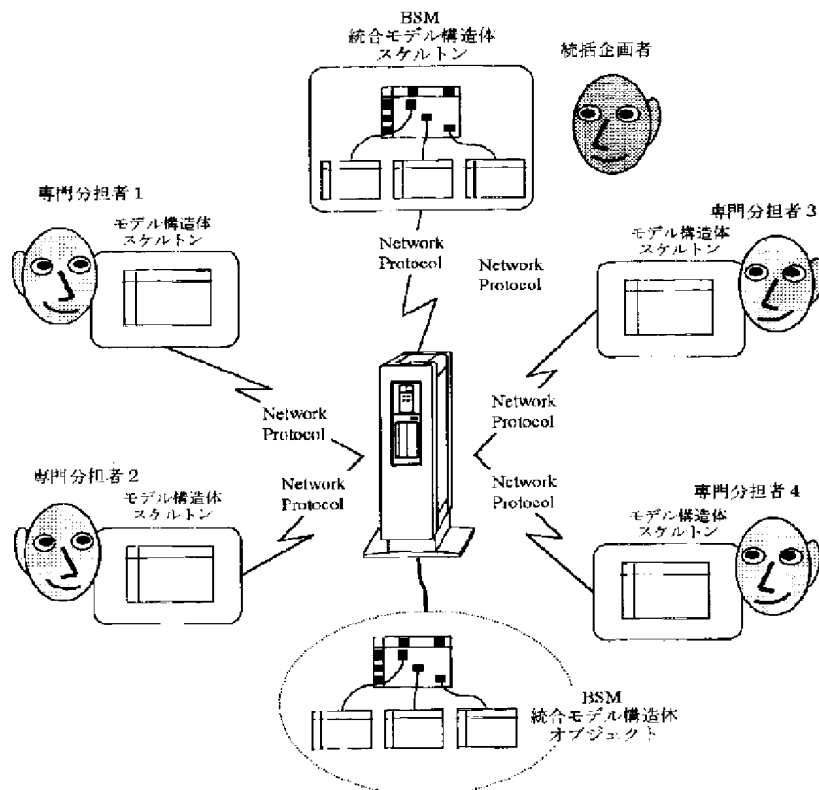


非有効域 個々を削除することでメインライン演算のシーケンスが確立可能

【図30】



【図31】



PAT-NO: JP02001357190A
DOCUMENT-IDENTIFIER: JP 2001357190 A
TITLE: METHOD REGARDING PLANNING,
DESIGNING, STRUCTURING,
MAINTENANCE, AND OPERATION
OF BSM
PUBN-DATE: December 26, 2001

INVENTOR-INFORMATION:

NAME	COUNTRY
HIGASHIHARA, ISAO	N/A
TOYAMA, TAKAYUKI	N/A

ASSIGNEE-INFORMATION:

NAME	COUNTRY
AVIX INC	N/A

APPL-NO: JP2001110877

APPL-DATE: April 10, 2001

PRIORITY-DATA: 2000111033 (April 12, 2000)

INT-CL (IPC): G06F017/60 , G06F019/00

ABSTRACT:

PROBLEM TO BE SOLVED: To provide a mechanism which grasps management and social events as a relation having 'round-robin cause and effect'

among their elements, performs mapping with a model composed of multi-layered, multidimensional structure and operated by an information processor, actualizes a modeling (BSM model) having bidirectional transparency enabling the reverse imaging of the events from the model with a table or a table group without depending upon an expression, and makes a cause and effect analysis of an intended condition and seeks a proper future behavior.

SOLUTION: This method has three areas for an input, a process, and an output and their elements form a nesting structure of a multi-layered housing based upon a housing having an operation relation and a sharing structure and data in various form are stored in rooms as storage places of the housing to form time-series models. Various links among the models are generated, and development operation and simulation by model operation enable end-user's interactive operation.

COPYRIGHT: (C) 2001, JPO